Time Domian Synthetic Aperture Radar Image Processing using Fast Factorised Backprojection

Shamiel Adams

A dissertation submitted to the Department of Electrical Engineering,
University of Cape Town, in fulfilment of the requirements
for the degree of Master of Science in Engineering.

Cape Town, 21 February 2005

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the degree of Master of Science in Engineering in the University of Cape Town. It has not been submitted before for any degree or examination in any other university.

Signature of Author ..... Signed by candidate .....

Cape Town

21 February 2005

Abstract

Fast Factorised Backprojection(FFBP) is a Synthetic Aperture Radar (SAR) focusing technique which uses Filtered Backprojection(FBP) to produce high quality images in the spatial domain at speeds which rival spectral domain SAR focusing methods.

Synthetic Aperture Radar (SAR) focusing is fundamentally a matched filtering process which can be performed both in the spatial or spectral domain. Spatial domain SAR focusing techniques like filtered backprojection can, in theory, completely recover the scene from data with an infinite bandwidth acquired using an isotropic antenna along an arbitrary curved flight path of infinite length.

This means that the focusing of a image using filtered backprojection is only limited by a finite bandwidth, non isotropic beam width and a finite flight path length. However filtered backprojection with an operation count which is proportional to $\mathcal{O}(n^3)$ is computationally inefficient in speed when compared to the frequency domain SAR image reconstruction algorithms with an operational count which is proportional to $\mathcal{O}(n^2 \log(n))$.

The FFBP combines the best of both worlds giving an algorithm which is suited to processing ultra wide band SAR data acquired along a arbitrary curved flight path using a radar that produces a wide beam while still having an instruction count which is proportional on to an $\mathcal{O}(n^2 \log(n))$ process.

This thesis describes the investigation of the fast backprojection focusing algorithm FFBP and compares the speed and quality of SAR inversion of the FFBP algorithm to that of the FBP algorithm using the same set of SAR data. The FFBP is explained here for an $L \times O$ range compressed unfocused SAR data set which is to be focused to a $M \times N$ pixel ground range image.

FFBP uses a divide and conquer approach to reduce the number of instructions needed to complete the processing task. A constant factor F is used to control the process of focusing through the S stages of processing. The main $M \times N$ pixel image is partitioned by a factor of $F \times F$ in each stage in a manner that produces $\frac{M}{F} \times \frac{N}{F}$ sub-images of $F \times F$ pixels in size in the final stage. Information is also gradually extracted from the $L \times O$ unfocused SAR data by reducing the number of aperture sample points in each stage by a factor of F to produce $\frac{M}{F} \times \frac{N}{F}$ sets of unfocused SAR data of size $F \times O$ to focusing each of the $F \times F$ sub-images.

It is shown in this thesis that the algorithm has a computational operation count proportional to $F \log_F(L) NM$ for the special case when the factor F, stages of processing S and number of aperture sampling points along the flight path L are related by $L = F^S$. The thesis also presents a derivation for the general case where the operation count is proportional to $\frac{SFLMN}{FS}$ irrespective of the relationship of F, S and L.

The FFBP is implemented here as a recursive, multi-stage SAR inversion algorithm. The SAR digital image reconstruction processor was designed to focus images to ground range and take

into account an arbitrary flight path, uneven ground and antenna beam pattern.

Tests showed that there is a vast increase in time saved when performing SAR inversion using the FFBP method. However the increase is most prevalent when ensuring that the number of aperture points, factor and number of stages used are related by $L = F^S$. Deviating from this theoretical optimum resulted in a deviation from the theoretically predicted saving.

The thesis project approaches the algorithm from a theoretical perspective. SAR inversion is performed on simulated unfocused basebanded range compressed SAR data which is representative of real SAR data. This SAR data set was produced by a simulator developed for the purpose of the thesis project, to provide raw SAR data for SAR inversion.

Mathematical models for filtered backprojection used for ground range and slant range focusing are presented. Also a radar echo model for representing the signal response of any point in the scene data acquired from an arbitrary flight path and beam pattern is derived.

Acknowledgements

I would like to acknowledge my supervisor Dr Andrew J Wilkinson for his assistance and help throughout the project and writing of this thesis.

To my family for their support, encouragement and patiences during this time.

The Defencetek branch of the CSIR in Tswane (Pretoria) who awarded the scholarship, financed by a SANDF, making this thesis possible.

The University of Cape Town and the department of Electrical Engineering for awarding the "Departmental Postgraduate Scholarship Bursary".

To my colleagues and staff in the Department of Electrical Engineering for their helpfulness.

Contents

List of Figures

List of Tables

List of Symbols

$A = A_{peak}V_t$	—	Arbitrary constant
A_{peak}	—	Peak amplitude output for convolution of matched filtered and design pulse
A_{ave}	—	Average amplitude
C_K	—	Proportionality constant relating received to transmitted voltage
c	—	Wave speed
E	—	Energy in waveform
df	—	Frequency resolution
dl	—	Flight path sample spacing
dx	—	Digital elevation map element spacing in x direction
dx_i	—	Ground reflectivity map element spacing in x direction
dx_m	—	Focused SAR data image pixel spacing in x direction
dR	—	Unfocused SAR data range bin spacing in range R
dt_A	—	Unfocused SAR data cross range bin spacing in time t_A
dy	—	Digital elevation map element spacing in y direction
dy_j	—	Ground reflectivity map element spacing in y direction
dy_n	—	Focused SAR data image pixel spacing in y direction
$d\theta_A$	—	Antenna beam patten element spacing in θ_A direction
$d\phi_E$	—	Antenna beam patten element spacing in ϕ_E direction
F	—	Constant factorisation factor
f	—	Frequency
f_c	—	Transmitted pulse centre frequency
f_f	—	Transmitted pulse final frequency
f_i	—	Transmitted pulse initial frequency
f_l	—	Constant used for aperture factorisation
f_m	—	Constant used for image partitioning in x direction
f_n	—	Constant used for image partitioning in y direction
f_S	—	Factorisation factor for final stage
f_s	—	General factorisation factor
$f\,(x,y)$, $f\,[m_x,n_y]$	—	Filtered Backprojection focused image
$g\,(t_A,R)$,$g\,[l,o]$	—	Modal of the receives range compressed unfocused SAR data at baseband
G, $G\,(\theta_A,\phi_E)$,$G\,[i_\theta,j_\phi]$	—	Antenna gain (Antenna beam patten)
G_r	—	Antenna Gain on reception
G_t	—	Antenna Gain on transmission
$h\,(x,y)$	—	Backprojection focused image

$H(x,y)$, $H[i_x, j_y]$	—	Height Map
I	—	Ground reflectivity map array element in x direction
I_A	—	Number of multi-buffer disk array sectors
$I_C(L,S)$, $I_C(L,S,F)$	—	FFBP instruction count
$I_F[s]$	—	Number of image partitions in x direction
$I_M[s]$	—	Number of image partitions in x direction per sub-image
$I_S(L,S)$, $I_S(L,S,F)$	—	FFBP Saving in comparison to FBP
I_X	—	Digital elevation map array size in x direction
I_Θ	—	Antenna beam patten array elements in θ_E direction
i	—	Indexing variable for ground reflectivity map in x direction
i_x	—	Indexing variable for digital elevation model map in x direction
$i_x(x)$	—	Digital Elevation Model cell index computed from x value.
i_f	—	Indexing variable for sub-images in y direction
i_θ	—	Indexing variable for antenna beam patten in θ_A direction
$i_\theta(\theta_A)$	—	Antenna beam patten cell index computed from θ_A value
J	—	Ground reflectivity map array elements in y direction
J_A	—	Number of elements in disk array sectors
$J_F[s]$	—	Number of image partitions in y direction
$J_N[s]$	—	Number of image partitions in y direction per sub-image
J_Y	—	Digital elevation map elements in y direction
J_Φ	—	Antenna beam patten array elements in ϕ_E direction
j	—	Indexing variable for ground reflectivity map in y direction
j_f	—	Sub image indexing variable j_f in y direction
j_y	—	Indexing variable for digital elevation model in y direction
$j_y(y)$	—	Digital elevation model cell index computed from y value
j_ϕ	—	Indexing variable for antenna beam patten in ϕ_E direction
$j_\phi(\phi_E)$	—	Antenna beam patten cell index computed from θ_E value
K_A	—	Total number of elements in multi-buffer disk array
$K_F[l_k]$	—	Array designating aperture sample point to sub aperture
$K_L[s]$	—	Number of new apertures points produced in a stage
k	—	Indexing variable of point targets
k_l	—	Variable indexing sub-apertures
L	—	Main aperture elements
L_A	—	Buffer length of multi-buffer disk array
L_F,$L_F[s]$, $L_F[s,k_l]$	—	Number of aperture points per sub-aperture
$L_K[s]$	—	Number of main aperture point in each stage
L_S	—	$L_S = F^S$
l	—	Main aperture indexing variable
l_k	—	Aperture indexing variable of newly formed apertures
M	—	Main image, pixels in x direction
M_A	—	Number of elements in 2D array (Dimension 1)
$M_F[s]$	—	Number of pixel in a sub-image in x direction
M_S	—	Maximum memory needed for data storage for processing
m	—	Main image indexing variable in x direction

N	—	Main image, pixels in y direction
N_A	—	Number of elements in 2D array (dimension 2)
$N_F[s]$	—	Number of pixel in a sub-image in y direction
n	—	Main image indexing variable in y direction
$n(f)$	—	Noise spectrum
$n(t)$	—	White noise
$n_o(t)$	—	Output of white noise from matched filter
O	—	Number of range bins in down range profile length of unfocused SAR data
O_A	—	Size of array storing memory buffers
$O_{MN}[s]$	—	Number of range bins processed by FFBP in each down range profile
$o_R(R[k,l])$	—	Unfocused SAR data bin index computed from $R[k,l]$ value.
$P(f)$	—	Spectrum of pulse centred around base banded
$P_{MF}(t)$	—	Matched filter spectrum of transmitted pulse centred around base banded
$P_{mf}(f)$	—	Matched filter spectrum of arbitrary pulse
P_r	—	Power received
P_t	—	Power transmitted
$p(t)$	—	Pulse centre around base banded
$p_c(t)$	—	Carrier signal
$p_{MF}(t)$	—	Impulse response of matched filtered
$p_{mf}(t)$	—	Matched filtered of pulse $p(t)$
$p_{RX}(t)$	—	Transmitted pulse
$p_{TX}(t)$	—	Received pulse
$p_o(t)$	—	Output of matched filtered in response to $p(t)$
$Q(f)$	—	Spectrum of matched filtered Output
$q(t)$	—	General match filtered output
$q(t_A, R, \vec{r})$	—	Phase history for single ideal point target having a flat frequency response
$q_{MF}(t)$	—	Range compressed output of single target of Downrange profile
$q_{MF}(t_A, R, \vec{r})$	—	Phase history of a single point target (continues)
$q_{MF}[l, o, \vec{r}[k]]$	—	Phase history of a single point target (discrete)
$q_{MF}(t_R)$	—	Range compressed output of single target of downrange profile
$q_{RX}(t)$	—	Down converted received pulse centred around baseband
R	—	Range
R_I	—	Unfocused SAR data Array a initial start range in R direction
$R[k,l]$	—	Range locus used in simulation of SAR data
$R[o]$	—	Unfocused SAR data bin centre of index o in range R
R_s	—	Slant range axis for focused image
$\vec{r}$	—	Position vector of point relative to the origin
$\vec{r}_i$	—	Position vector of a point on the flight path
$\vec{r}_f$	—	Position vector of a point target in the scene
$\vec{r}[k], \vec{r}[i,j]$	—	Position vector map of point targets in the scene
S	—	FFBP processing stages
S_F	—	Maximum number of stages possible for processing
S_L	—	Maximum number of stages possible for processing limited by aperture size
S_M	—	Memory saving ratio

S_{MN}	—	Maximum number of stages possible for processing limited by image size
s	—	FFBP stage indexing variable
$\vec{s_R}$	—	Range vector
$\vec{s_u}\left[s, k_l, l_k\right]$	—	Vector function representing a sub-aperture sampling points
s_x,s_y,s_z	—	Range vector distance in x, y, z direction
T_I	—	Unfocused SAR data Array azimuth start time in t_A direction
t	—	Time
t_A	—	Azimuth time along flight path
$t_A\left[l\right]$	—	Unfocused SAR data bin centre of index l in time t_A
t_i	—	Initial time at start of integration
t_f	—	Final time at the end of integration
t_R	—	Time in range direction
t_s	—	Zero doppler time for focused image
u	—	Nominal flight path axis
$\vec{u}\left(t_A\right)$, $\vec{u}\left[l\right]$	—	Radar flight path
$\vec{u}_s\left[s, k_l F + l_k\right]$	—	Vector function representing aperture sampling points
$u_x\left(t_A\right)$, $u_y\left(t_A\right)$, $u_z\left(t_A\right)$	—	Radar flight path distance in x, y and z direction
V_r	—	Received voltage
V_t	—	Transmitted Voltage
X_C	—	Focused SAR data (main image) image centre in x direction
X_I	—	Digital elevation map centre in x direction
X_σ	—	Ground reflectivity map centre in x direction
x	—	Distance along x axes
$\hat{\mathbf{x}}$	—	Positive unit vector in $x$direction
$x\left[i_x\right]$	—	Digital elevation model cell centre of index i_x in x direction
$x\left[k\right]$	—	Distance map of point targets in the scene in x direction
$x_i\left[i\right]$	—	Ground reflectivity map cell centre of index i in $x$direction
$x_m\left[m\right]$	—	Focused SAR data pixel centre of index m in x direction
x_0	—	Location of Point target along the x axis
Y_C	—	Focused SAR data (main image) image centre in y direction
Y_J	—	Digital elevation map centre in y direction
Y_σ	—	Ground reflectivity map centre in y direction
y	—	Distance along y axes
$\hat{\mathbf{y}}$	—	Positive unit vector in $x$direction
y_0	—	Location of point target along the y axis
$y_j\left[j\right]$	—	Ground reflectivity map cell centre of index j in y direction
$y\left[j_y\right]$	—	Digital elevation model cell centre index j_y in y direction
$y\left[k\right]$	—	Distance map of point targets in the scene in y direction
$y_n\left[n\right]$	—	Focused SAR data pixel centre of index n in y direction
Z_c	—	Digital elevation map centre in z direction
z	—	Distance along z axes
$\hat{\mathbf{z}}$	—	Positive unit vector in x direction
$z\left[k\right]$	—	Distance map of point targets in the scene in z direction
Δf	—	Pulse bandwidth

ΔR	—	Unfocused SAR data array down range profile length in R direction
Δt	—	Sweep time foe chirp pulse
Δt_A	—	Unfocused SAR data array cross range length in time length in t_A direction
$\Delta \kappa$	—	Spacial bandwidth of transmitted pulse
Δx	—	Digital elevation map length in x direction
Δx_i	—	Ground reflectivity map length in x direction
Δx_m	—	Focused SAR data Image length in $x$direction
Δy	—	Digital elevation map length in y direction
Δy_j	—	Ground reflectivity map length in y direction
Δy_n	—	Focused SAR data Image length in y direction
$\Delta \theta_A$	—	Antennae beam patten angular length in θ_A direction
$\Delta \upsilon$	—	Main lobe beam width
$\Delta \phi_E$	—	Antennae beam patten angular length in ϕ_E direction
δA	—	Ultimate resolving power for ultra wideband, widebeam SAR moving along a linea
δR	—	Maximum possible focused resolution in range
δl	—	Maximum possible focused resolution in azimuth
ζ,$\zeta\,(\vec{r})$,$\zeta\,[i,j]$	—	Reflectivity distribution
$\frac{\eta}{2}$	—	Average noise of noise spectrum
Θ_A	—	Antenna beam patten centre in θ_A direction
θ_A	—	Azimuth angle from radar boresight
$\theta_A[i_\theta]$	—	Antenna beam patten cell centre in $\theta_A$direction
θ_R	—	Squint angle of vector s_R
θ_{Squint}	—	Squint angle of radar
θ_z	—	Angle between vector $\langle x, y, 0\rangle$and x axis
κ	—	Spacial frequency in range
κ_c	—	Spacial carrier frequency
κ_x	—	Spacial frequency along x axis
λ	—	Wave length
λ_c	—	Wave length of centre frequency
λ_i	—	Wave length of final frequency
λ_f	—	Wave length of inital frequency
ρ	—	Distance in spherical coordinates form origin to point in 3D space
ρ_z	—	Radial distance from origin on $x-y$ plane
σ	—	Radar cross section
τ	—	Time delay of returning reflected pulse
Φ_E	—	Antenna beam patten cell centre of index i_θ in $\theta_A$direction
ϕ_E	—	Elevation angle from radar boresight
$\phi_E[j_\phi]$	—	Antenna beam patten cell centre of index j_ϕ in $\phi_E$direction
ϕ_R	—	Depression angle of vector s_R
$\phi_{Depression}$	—	Depression angle of radar
ϕ_z	—	Angle between vector $\langle x, y, 0\rangle$and vector $\langle x, y, z\rangle$

Nomenclature

Azimuth—Angle in a horizontal plane, relative to a fixed reference, usually north or the longitudinal reference axis of the aircraft or satellite.

Backprojecting—Backprojection is one of the methods for computing the inverse of the Radon transform by reconstructing the image from its Radon projection.

Beamwidth—The angular width of a slice through the mainlobe of the radiation pattern of an antenna in the horizontal, vertical or other plane.

Matlab—A scientific programing language used mostly for design and simulation.

PRF—Pulse repetition frequency.

Range—The radial distance from a radar to a target.

Steradian— The unit of solid or three dimensional angle

Synthetic Aperture Radar (SAR)—A signal-processing technique for improving the azimuth resolution beyond the beamwidth of the physical antenna actually used in the radar system. This is done by synthesising the equivalent of a very long side looking array antenna.

Polymorphism— In object oriented programing: the ability if an object to hold objects of several different types. When operations are applied to a polymorphic type, the operation which is appropriate to the actual type is automatically selected [21].

Acronyms

ABP	—	Antenna Beam Patten
CARABAS	—	Coherent All RAdio BAnd Sensing
DFT	—	Discrete Fourier Transform
FBP	—	Filtered Backprojection
FFBP	—	Fast Filtered Backprojection
FFT	—	Fast Fourier Transform
FOA	—	Swedish Defence Agency
FSD	—	Focused SAR Data
GRM	—	Ground Reflectivity Map
LBP	—	Local Backprojection
PRF	—	Pulse Repetition Frequency
PRI	—	Pulse Repetition Interval
RCS	—	Radar Cross Section
RFP	—	Radar Flight Path
SAR	—	Synthetic Aperture Radar
SAS	—	Synthetic Aperture Sonar
TDC	—	Time Domain Correlation
TSW	—	Transmitting Signal Waveform
UCT	—	University of Cape Town
USD	—	Unfocused SAR data
UWB	—	Ultra Wide Band
WB	—	Wide Beam
jpeg	—	joint photographic experts group
pgm	—	pixel grey map
ps	—	post script

Chapter 1

Introduction

SAR focusing is fundamentally a correlation process which can be performed both in the spatial or spectral domain. Techniques such as *time domain correlation* (TDC) [14] and Backprojection [14] which use convolution techniques in the spatial domain give better digitally reconstructed images than their spectral domain counterparts [14]. However these procedures are computationally inefficient in speed in comparison to the frequency domain SAR image reconstruction algorithms. The speed of image focusing is greatly increased for frequency domain processing by two important factors. The time consuming, computationally intensive, process of spatial domain convolution, is a straight forward multiplication process in the spectral domain and the speed at which the Fast Fourier Transform (FFT), transforms data between the spatial and spectral domain also ensures faster data throughput[1]. However with this increase in speed comes a degradation in quality of the final focused image, as a result of approximations made especially if the aperture is curved and unevenly sampled (non-linear aperture) [1].

Producing a synthetic aperture where sampling points are evenly spaced on a straight line is the ideal scenario for exact SAR inversion. This is referred to as a linear aperture by [1]. However, a linear aperture is not the general case as the synthetic aperture is generally unevenly sampled and curved. This is referred to as a non-linear aperture [1]. According to [1] exact inversion of a non-linear aperture is possible in the time domain by using a generalisation of filtered backprojection as an inversion method as used in the field of tomography. The detailed theory of backprojection can be found in [6] and [7]. The type of backprojection which performs exact inversion used in [1] and [2] is known as Filtered Backprojection(FBP) and the theory which explains this method can be found in [15].

It is not always possible to achieve exact SAR inversion using frequency domain SAR focusing techniques. In recent years researchers developed algorithms such as the "Range Migration"[1, 5, 4] and "Fourier-Hankel"[1, 9, 10] which perform exact inversion for linear apertures in the frequency domain [1]. It is not easy to extend these algorithms to handle exact SAR inversion for non-linear apertures [1]. For SAR inversion in the frequency domain, [1] state "It is possible to partly correct for small non-linear motion by dividing the image into small sub-images which need to be processed separately since motion compensation is spatially variant". However, this does not perform an exact inversion. According to [1] the motion problem becomes a major issue

[1]2D spatial domain convolution has an operation count which is proportional to an $\mathcal{O}\left(n^4\right)$ process. On the other hand a 2D multiplication process has a operation count proportional to $\mathcal{O}\left(n^2\right)$, and the 2D FFT has a operation count which is proportional to $\mathcal{O}\left(n^2 2\log n\right)$ process [20] . Thus the SAR processing in the spatial domain has at best an operation count which is proportional to an $\mathcal{O}\left(n^2 2\log n\right)$ process.

when trying to achieve an acceptable focused resolution when processing Wide-Beam (WB) SAR using frequency domain techniques. Also in [11] it is stated that it can be shown that in the case of a deviation from a linear aperture, using frequency domain techniques seriously degrades the focused image in Synthetic Aperture Sonar (SAS).

Although time domain filtered backprojection SAR provides an exact inversion for a general non-linear aperture its main drawback is the time consuming nature of the process. Inversion of a SAR scene containing L aperture points which must be focused to an image of dimension $M \times N$ requires an operation count which is proportional to $M \times N \times L$. If the image dimensions are equal to the aperture length $M = L$ and $N = L$ it can be seen that the operation count is proportional to $L \times L \times L = L^3$. Thus the number of operations required to focus a SAR scene has an operation count which is proportional to an $\mathcal{O}(n^3)$ process [11].

Frequency domain SAR inversion uses the FFT to speed up the time consuming process of focusing. In one dimension the FFT has an operation count which is proportional to an $\mathcal{O}(n \log n)$ [20] process. Using a 2D FFT on two dimensional data such as SAR data gives an operation count which is proportional to an $\mathbf{O}(n^2 \log n^2) = \mathcal{O}(n^2 2 \log n)$ process [20]. By just using the FFT as a tool for SAR inversion a saving which is at best $\frac{n}{2 \log n}$ times faster than the normal time domain processing could be achieved. Currently, frequency domain SAR inversion is only exact for a linear aperture using algorithms like the Range Migration and Fourier-Hankel which use the FFT to speed up the focusing process [1].

Both [1] and [2] stress the need for new fast processing algorithms which can be adapted for inversion of a general curved aperture geometry. It is also suggested by [1] that the most obvious solution is to produce fast algorithms in the time domain. This makes sense since [1, 11, 6] all claim exact inversion for focusing SAR data for a nonlinear aperture in comparison to frequency domain techniques where exact inversion methods of recent algorithms require a linear aperture.

Recently there has been a number of new suggested fast algorithm methodologies which speed up SAR inversion using filtered backprojection in the time domain [1] which give theoretical operation counts proportional to an $\mathcal{O}(n^2 2 \log n)$ process. These are the "quad-tree"[1], "image tessellation"[1] and "aperture factorisation" [1, 2] focusing methods [11]. These fast time domain SAR inversion techniques using filtered backprojection are derived from an exact inversion method and are thus better suited to focus images from nonlinear apertures than their fast frequency domain counterparts for high quality SAR inversion.

This thesis project focused on implementing time domain image processing using aperture factorisation. The thesis project has implemented the fast time domain inversion technique found in [1] and [2] called Fast Factorised Backprojection (FFBP).

FFBP was developed by the Swedish Defence Agency (FOA) in Sweden and is a fast time domain, ultra wide band (UWB) focusing algorithm [1, 2] which is capable of inversion of raw SAR data which is acquired along a nonlinear flight path using a radar with an antenna which produces a wide beamwidth. The FOA version operates on low VHF or UHF frequency data [1, 2]. The algorithm focuses images by processing the data through a number of stages which can be specified. Theoretically the algorithm focuses images faster when using more stages of processing. However the quality of the image degrades as more stages are used. The one stage version of the FFBP algorithm is the normal filtered backprojection algorithm [1, 2].

Another fast algorithm which uses aperture factorisation for SAR inversion is referred to as Local Backprojection (LBP) [3]. This algorithm has a theoretical operation count proportional to an

$\mathcal{O}\left(n^{2.5}\right)$ and is computationally equivalent to the FFBP algorithm using 2 stages [2]. Both the LBP and FFBP algorithms will be analysed in this thesis.

The FFBP algorithm has been investigated because it promises high quality images at speeds which rival frequency domain SAR techniques from SAR data captured by ultra wide band, wide beam SAR system along a nonlinear flight path.

1.1 Thesis Motivation, Objective and Scope

Since [1] stated that exact inversion of a non-linear aperture is possible in the time domain by using a generalisation of filtered backprojection as an inversion methodology, the FFBP algorithm was chosen because the above statement implies that time domain filtered backprojection is an exact inversion method for the theoretical case of SAR focusing of a 2-dimensional scene using 2-dimensional unfocused SAR data that were acquired along a 1-dimensional non-linear aperture of infinite length by a radar which has an infinite bandwidth and produces an isotropic antenna.[2] The theory for frequency domain algorithms at present is derived from methods which are not easily extended to processing unfocused SAR data acquired from a non-linear flight path. Since the theory of filtered backprojection is suited to a general flight path the FFBP can focus images accurately with minimum amount of distortion while still competing with frequency domain methods. Thus FFBP is of particular interest since it combines the accuracy of time domain focusing with the speed of fast algorithms.

The thesis project objective was to investigate the Multi-stage Fast Factorised Filter Backprojection SAR inversion method, FFBP which is explained in [1] and [2]. The aim was to use this method to implement a program which is capable of processing unfocused basebanded range compressed SAR data to produce a focused ground range image with the aid of a flight path, digital elevation model and a map of the antenna beam patten using the FFBP method. A comparative study is done by comparing time taken to process an image using FBP to the time taken to process the same image using FFBP. This thesis will report on the efficiency of the time saving capability of the FFBP implementation in this thesis project under different processing scenarios.

The thesis project approaches the algorithm from a theoretical perspective. Thus this thesis will not concentrate on focusing real SAR data but on SAR inversion of simulated unfocused basebanded range compressed SAR data instead. For this reason a detailed model of received basebanded SAR data is derived inorder to produce data which resembles real SAR data. The thesis is strictly concerned with the processing of SAR data and does not look at issues associated with acquiring the data. Only relevant signal processing techniques in data acquisition are dealt with for the purpose of implementing an accurate radar echo model for unfocused base banded range compressed data that is comparable to data acquired from a real scene.

1.2 Contents of Thesis

Chapter 2 deals with preliminary theory needed to explain concepts throughout the thesis. The chapter starts off by discussing, describing and defining SAR modes, SAR geometry and the co-

[2] time domain filtered backprojection is an exact inversion method for the theoretical case of SAR focusing of a $N+1$ dimensional scene using $N+1$ dimensional unfocused SAR data that was acquired along a N dimensional non-linear aperture of infinite length by a radar which has an infinite bandwidth and utilises an isotropic antenna.

ordinate systems used to explain concepts in this report. Then the modelling of the radar flight path, antenna beam patten and recorded voltage based on the radar equation used for explanation throughout the thesis is also covered here. An explanation of the process of matched filtering, used to implement the range compressed radar echo model, is also given. The chapter also builds a radar echo model of the received phase history for unfocused basebanded range compressed SAR data. This model is used by the simulator which produces the unfocused SAR data for inversion by the SAR focusing algorithms.

Chapter 3 deals with the theory of backprojection used for SAR inversion. It starts by giving a brief history of backprojection and the *Radon Transform*. Then backprojection is discussed in the context of SAR inversion. The chapter shows that, in theory, using filtered backprojection for SAR focusing is an exact inversion method. Models for filtered backprojection used for ground range and slant range focusing are presented.

Chapter 4 describes the digital image reconstruction method FBP and FFBP algorithm. Firstly one stage backprojection (FBP) is described and shown to be proportional to an $\mathcal{O}(n^3)$ process. Then the fast backprojection algorithm (FFBP) used to reduce the amount of processing time needed to digitally reconstruct an image is explained. The partitioning of images, partitioning of apertures, program recursion and computational complexity of the algorithms are described. Also discussed are the theoretical instruction count and theoretical efficiency of the algorithms.

Chapter 5 discusses the actual program implementation of the algorithms. Detailed flow diagrams are provided to illustrate the method of implementation. This chapter also addresses the practical versus the theoretical savings of the algorithm which is derived in chapter 4.2. The reasons for porting of the FFBP algorithm from Matlab to C++ and the advantages thereof are discussed. The practicalities of data movement in a physical PC which slow the FFBP algorithm down from its theoretical efficiency are also discussed here.

Chapter 6 reports on the results obtained from testing the speed of the normal one-stage filter backprojection algorithm in comparison to the multistage fast factorised filter backprojection algorithm.

Chapter 7 draws conclusion and makes a few recommendations.

Appendix A gives a mathematical description of the Radon transform as used in medical image processing in two dimensions.

Chapter 2

Radar Echo Model

The investigation of the fast backprojection focusing algorithm compares the speed and quality of SAR inversion of the FFBP algorithms to that of the FBP algorithm using the same set of SAR data. The various focusing algorithms like the FBP, LBP and FFBP which were implemented all use simulated range compressed SAR data for SAR inversion. These SAR data were produced by a simulator developed for the purpose of the thesis project, to provide raw SAR data for SAR inversion. The details of the simulator implementation is given in Section 5.3. This chapter describes the derivation of the radar echo model which was used to design and implement the simulator that provided the range compressed SAR data. The SAR focusing algorithms described in this thesis were also implemented for SAR inversion of this radar echo model.

The backprojection SAR focusing algorithms discussed in [1] and [2] are UWB focusing algorithms [1, 2] which are capable of inversion of raw SAR data which are acquired from a 3D uneven ground as well as from a nonlinear flight path using a radar with an antenna which produces a wide beamwidth operating at low VHF or UHF frequencies. Thus the radar echo model must be able to represent SAR data which have all of the above characteristics.

For simplification the radar echo model will be representative of the received range compressed SAR data, which is raw SAR data which is translated down in frequency to baseband and then matched filtered in range. Thus only the bandwidth and centre frequency of the transmitted pulse are needed for the derivation of this range compressed radar echo model. The model must be able to predict:

1. Data which are acquired from a nonlinear aperture (i.e. curved trajectory),
2. The signature of range compressed SAR data reflected from any point in a 3D scene, and;
3. The effect of any antenna beam pattern.

As in [1] and [2] a few assumptions are made for the derivation of the radar echo model. It is assumed that the model will be derived for:

1. Raw SAR data acquired by a monostatic radar - i.e. the receiving and transmitting antenna are collocated [1],
2. A linear propagation medium which is isotropic, homogeneous and non-dispersive - i.e. the wave velocity c is constant throughout the medium [1],

3. A radar which is stationary while transmitting and receiving data - NB. For real SAR data acquisition, the speed of electromagnetic radiation of the transmitted pulse is much faster than that of the radar moving along its flight path. Thus simulating in real SAR it is assumed that the radar is stationary. This is commonly referred to as the start stop approximation [14].

4. A ground consisting of a collection of single-scattering objects (Born approximation) [1, 2], also referred to as point targets in this thesis, whose received signatures can be summed up (superposition) to represent the complete radar echo model .

This chapter starts by describing SAR modes, SAR geometry and the co-ordinate systems used in deriving the radar echo model. The modelling of the radar flight path and antenna beam pattern are then discussed. The modelling of the recorded voltage, due to the received average power of a returning signal, is based on the radar equation. A brief explanation of the process of matched filtering used to implement the radar echo model is also described.

The radar echo model must represent the recorded basebanded range compressed SAR data. For the purpose of showing how the radar echo model is derived, a detailed step by step derivation of a single point target from pulse transmission to data reception, baseband conversion and matched filtering is described for a single downrange profile. Then, this single downrange profile is extended to the 2D case where the phase history of all downrange profiles for a single point target is represented. The complete radar echo model, representing multiple target signatures, is then derived by the summation of all 2D point target signatures.

Since this thesis is concerned mainly with SAR focusing, the radar echo model is a simple representation of received data and ignores the following:

- the effects of layover and shadowing
- the information about the transmitted pulse; such as pulse length or spectral components.
- The effect of polarisation due to the antenna beam pattern

2.1 SAR Modes

The ways in which the radar moves and illuminates the ground are referred to as Synthetic Aperture Radar Modes. Three common modes used for synthetic aperture data capturing are [4].

1. Strip Map Synthetic Aperture Radar [4].
2. Spotlight Synthetic Aperture Radar [4].
3. Scan Synthetic Aperture Radar [4].

In all of the above modes of SAR the moving platform will attempt to travel in a straight line with its beam pointing downward at an angle towards the ground. The radar beam is also *side looking* [4] (i.e. the beam will always be pointed towards one side of the moving platform; and never to wards an angle in the direction of the flight path either in front of or behind the moving platform).

- Strip Map SAR is achieved by the radar having a *side looking* beam which is not steered as the radar platform moves and illuminates the ground.

- Spotlight SAR is achieved by the radar continuously illuminating the same surface of the ground. This would require a radar mounted on a moving platform to steer the beam in azimuth only so that it always illuminates its desired target area.
- Scan SAR is a combination of Strip Map and Spotlight SAR. The *side looking* beam can be steered in elevation as well as azimuth so that it can illuminate any area along track.

2.1.1 Strip Map SAR Geometry

The *side looking* strip map geometry used for the explanation in this report is given in Figure 2.1. The radar platform moves along a flight path above the ground, with its antenna beam looking to the side, illuminating the terrain under investigation. Ideally the radar platform should move along a straight line at a constant speed.

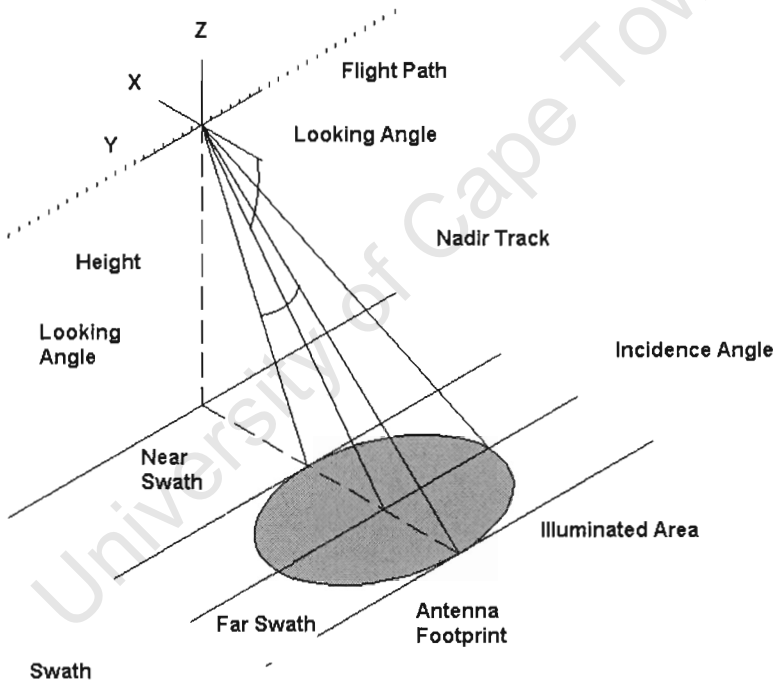


Figure 2.1: SAR strip map geometry

The ideal straight path which the radar should travel on is known as the *nominal flight path*. The trace on the ground vertically under the nominal flight path is known as the *nadir track*. The angle at which the radar is pointed to the ground is referred to as the *depression angle* and the illuminated surface is known as the *footprint* of the radar. The track which is illuminated over time is called the *radar swath*. The near and far swath are the imaginary lines traced out by the nearest and furthest part of the illuminating footprint.

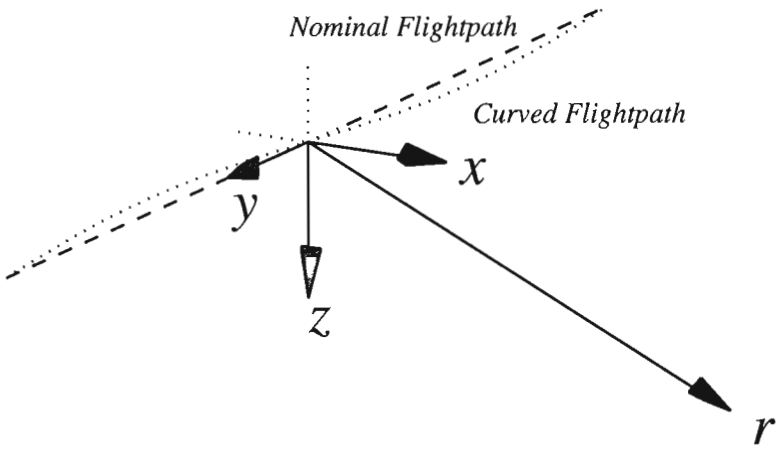


Figure 2.2: Right handed SAR Cartesian coordinate system

2.1.2 Co-ordinate System

The strip map SAR geometry described above uses a Cartesian co-ordinate system (x, y, z). Figure 2.2 shows the positive unit vectors designating the coordinate system, where $\hat{\mathbf{x}} = \langle 1, 0, 0 \rangle$,$\hat{\mathbf{y}} = \langle 0, 1, 0 \rangle$ and $\hat{\mathbf{z}} = \langle 0, 0, 1 \rangle$. The nominal radar flight path is aligned with the y axis. The z axis points vertically down wards to the nadir track. The x axis is defined by the cross product of the y and z axis $\hat{x} = \hat{y} \times \hat{z}$. The $\hat{\mathbf{y}}$ direction is also referred to as the *along track* and the $\hat{\mathbf{x}}$ direction is referred to the *across track*.

The vector representing a point relative to the origin is given by Cartesian co-ordinates as:

$$\vec{r} = \langle x, y, z \rangle \tag{2.1}$$
$$= x\hat{\mathbf{x}} + y\hat{\mathbf{y}} + z\hat{\mathbf{z}} \tag{2.2}$$

where x, y and z are scalar variables representing distance in the $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$ directions respectively. The length of the vector $\vec{r}$ is given by:

$$\rho = |\vec{r}| \tag{2.3}$$

Other quantities use to explain concepts in this report and shown in Figure 2.3 are:

- The distance to the projected point $(x, y, 0)$ of the vector $\vec{r}$ onto the $x - y$ plane given by:

$$\rho_z = \sqrt{x^2 + y^2} \tag{2.4}$$
$$= |\langle x, y, 0 \rangle| \tag{2.5}$$

- The angle that the vector $\langle x, y, 0 \rangle$ makes with the vector $\hat{\mathbf{x}}$ on the $x - y$ plane given by:

$$\theta_z = \arg(y, x) \tag{2.6}$$

- The angle that the vector $\langle x, y, 0 \rangle$ makes with the vector $\langle x, y, z \rangle$ given by:

$$\phi_z = \arctan(z, \rho_z) \tag{2.7}$$

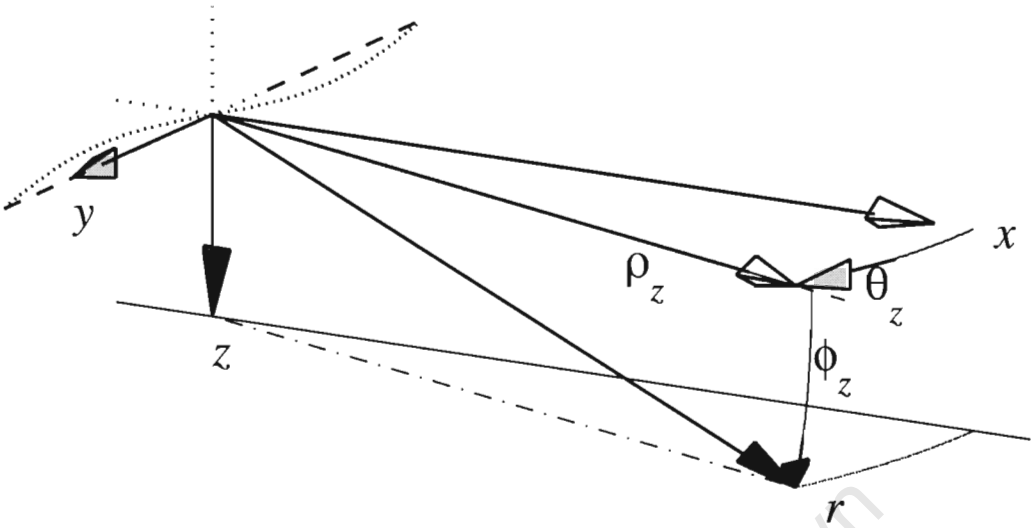


Figure 2.3: Additional Spherical and Cylindrical co-ordinates

2.2 Flight Path Modelling

Raw synthetic aperture data are acquired as the sending (transmitting) and receiving (recording) sensors are moved along a path (e.g. straight line). The modelling of flight path for SAR data acquisition is represented by the vectored parametric equation:

$$\vec{u}(t_A) = \langle u_x(t_A), u_y(t_A), u_z(t_A) \rangle \tag{2.8}$$

where $u_x(t_A)$, $u_y(t_A)$ and $u_z(t_A)$ are scalar functions representing distance in the $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$ directions and t_A is the controlling variable of the parametric equation representing the time along the flight path. Figure 2.4 shows a curved radar flight which can be represented by equation 2.8. The trace of the individual scalar functions $u_x(t_A)$, $u_y(t_A)$ and $u_z(t_A)$ on their respective $y-z$, $x-z$ and $x-y$ planes are given, also shown in below Figure 2.4 .

As the radar moves along the flight path it illuminates its target area with a pulse and then records the echoes which are reflected from various targets in the scene over a period of time. The pulses are transmitted at constant intervals in time as the radar moves along its path [13, 14]. The time period between successive transmitted pulses is referred to as the *pulse repetition interval* (PRI) [14]. The frequency of a PRI is called the *pulse repetition frequency* (PRF) [14].

In-between pulses the radar will record the returning pulses from the targets in the illuminated scene. The independent variable measuring the time delay for a returning pulse is given by t_R. The variable t_R always has the value of zero seconds at the start of every pulse transmission. Since the wave speed has a constant velocity the time for the round trip of the returning pulse is directly related to the distance of the target from the radar, given by

$$t_R = \frac{2R}{c} \tag{2.9}$$

[13, 14]. Therefore the range is given by:

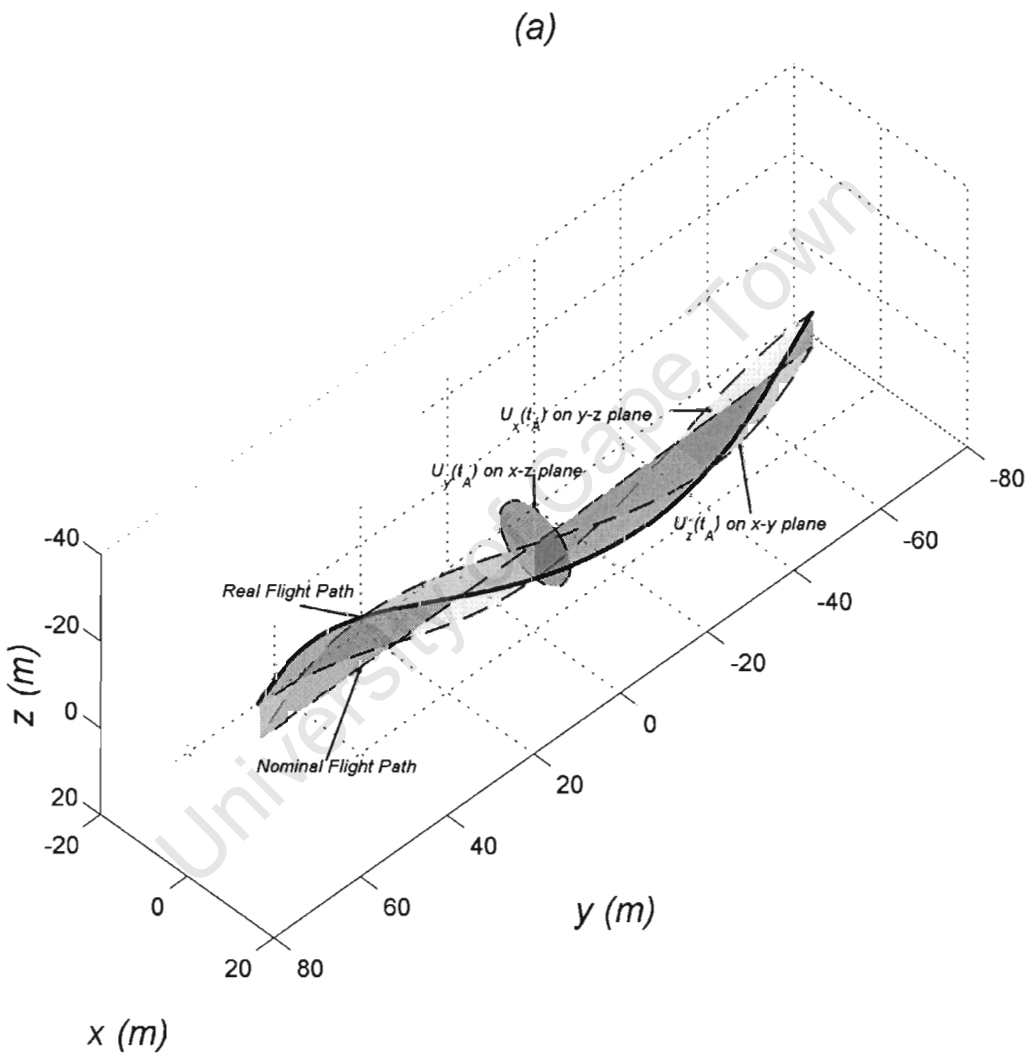


Figure 2.4: Radar flight path

The diagram shows a nonlinear flight path with the traced of the function $u_x(t_A)$, $u_y(t_A)$ and $u_z(t_A)$on the $y-z$, $x-z$ and $x-y$ planes respectively.

$$R = \frac{t_R c}{2} \tag{2.10}$$

The variables R and t_A represent the co-ordinates used in model and represent the axes of the unfocused range compressed SAR data. Since the flight path can represent a nonlinear aperture the variable t_A is a the time control variable of the parametric equation 2.8 measured in seconds and is in no way directly related to the flight path position at any point in time. The variable R is measured in metres and gives the distance which is directly related to time t_R by a factor of $\frac{c}{2}$.

The range of a target is defined here as the line of sight distance a signal has to travel from antenna to a target positioned in a 3D scene. Range to a point in a 3D scene in this radar echo model is given by :

$$R = |\vec{r} - \vec{u}(t_A)| \tag{2.11}$$

where $\vec{r}$ represents a point in the scene, $u(\vec{t}_A)$ represents the position of the radar in the flight path and R represents the range from the point $u(\vec{t}_A)$ to the point $\vec{r}$. The above relation will be referred to as the range locus in this thesis and gives the range R to a target at a point t_A along the flight path. The vector quantity $\vec{r} - u(\vec{t}_A)$ will be referred to in this thesis as the range vector and is given by :

$$\vec{s_R} = \vec{r} - \vec{u}(t_A) \tag{2.12}$$

Thus the range is also given by:

$$R = |\vec{s}_R| \tag{2.13}$$

$$= |\langle s_x, s_y, s_z \rangle| \tag{2.14}$$

where s_x,s_y and s_z are scalars representing the vector distance from antenna to target in the $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$ directions, respectively.

2.3 Antenna Beam Pattern

In the radar echo model of [1] and [2] some of the scaling factors like the response of the antenna and target objects are neglected because these are complex functions of frequency, polarisation and direction. The antenna model used in this thesis only represents the scaling response of a signal due to direction.

The effect of the antenna gain on a signal is modelled by using a distribution $G(\theta_A, \phi_E)$ which is dependent on the angles θ_A and ϕ_E which the target makes with the radar boresight. The angle θ_A represents the azimuth angle and the angle ϕ_E represents the elevation angle of the radar. There is no functional expression given to $G(\theta_A, \phi_E)$ in the radar echo model. The distribution $G(\theta_A, \phi_E)$ depends on the antenna beam pattern being provided used to model received data. The map $G(\theta_A, \phi_E)$ represents the way in which the antenna amplifies the target

response distributed over its 4π steradian. The total area underneath the radar beam pattern is $\int_{-\pi}^{\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} G(\theta_A, \phi_E)\, d\theta_A d\phi_E = 4\pi$.

Figure 2.5 shows a distribution that approximates a 30 degree beam in azimuth and elevation.

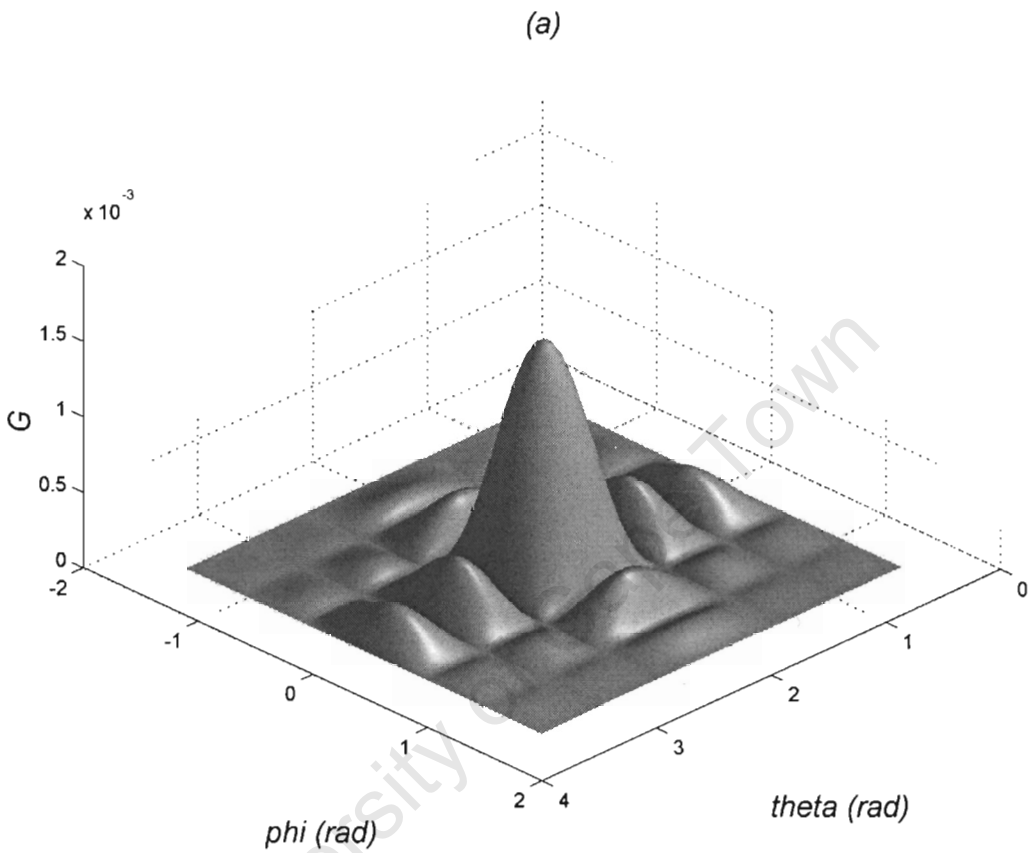


Figure 2.5: Antenna beam pattern

2.3.1 Modelling the gain

The squint angle θ_{squint} and azimuth angle θ_A are defined in the same manner as the angle θ_z. Likewise the depression angle $\phi_{depression}$ and elevation ϕ_E angle are also defined in the same manner as the ϕ_z angle .

The angles θ_A and ϕ_E used to index the distribution $G(\theta_A, \phi_E)$ are variables which related the angular position of the radar antenna $\vec{u}(t_A)$ to the target position $\vec{r}$.

- This section describes the calculation of the angles θ_A and ϕ_E for a radar where the boresight points in the $\hat{\mathbf{x}}$ direction.
- Then the calculation is extended in the next section to include the effects of the depression angle $\phi_{depression}$ and squint angle θ_{squint}.

For the special situation where the radar is positioned at the origin and where the radar boresight is aligned with the x axis, the angles θ_A and ϕ_E are exactly equal to θ_z and ϕ_z

$$\theta_A = \theta_z \tag{2.15}$$
$$\phi_E = \phi_z \tag{2.16}$$

For any general flight path where the radar boresight is always pointing in the $\hat{\mathbf{x}}$ direction the angles used to determine the gain can be found from the range vector $\vec{s}_R$ given by:

$$\theta_R(\vec{s}_R) = \arg(s_y, s_x) \tag{2.17}$$
$$\phi_R(\vec{s}_R) = \arctan\left(s_z, \sqrt{s_y^2 + s_x^2}\right) \tag{2.18}$$

where $\vec{s}_R = \vec{r} - \vec{u}(t_A)$ is the range vector, θ_R is squint angle and ϕ_R is depression angle of the vector $\vec{s}_R$.

Thus:

$$\theta_A = \theta_R(\vec{s}_R) \tag{2.19}$$
$$\phi_E = \phi_R(\vec{s}_R) \tag{2.20}$$

2.3.1.1 Antenna Squint and Depression Angle

In the radar echo model, the squint and depression angles of the radar have to be accounted for in order to calculate the proper azimuth and elevation angles. Thus the azimuth and elevation angle are given by taking the difference of the squint and depression angles of the vector $\vec{s}_R$ and the radar antenna given by:

$$\theta_A(\vec{s}_R) = \theta_R(\vec{s}_R) - \theta_{squint} \tag{2.21}$$
$$\phi_E(\vec{s}_R) = \phi_R(\vec{s}_R) - \phi_{depression} \tag{2.22}$$

2.3.2 Gain using vector

Since the squint and depression angles are found using the range vector $\vec{s}_R$ the gain in this radar echo model is also modelled using the vector $\vec{r}$ and $\vec{u}(t_A)$. The azimuth and elevation angle of the radar can be given by:

$$\theta_A(\vec{r}, \vec{u}(t_A)) = \theta_R(\vec{r}, \vec{u}(t_A)) - \theta_{squint} \tag{2.23}$$
$$\phi_E(\vec{r}, \vec{u}(t_A)) = \phi_R(\vec{r}, \vec{u}(t_A)) - \phi_{depression} \tag{2.24}$$

Thus the gain G is dependent on the angles $\theta_A(\vec{s})$ and $\phi_E(\vec{s})$ which can be determined by the vectors $\vec{r}$ and $\vec{u}(t_A)$ as shown below:

$$G = G(\theta_A(\vec{s}_R), \phi_E(\vec{s}_R)) \tag{2.25}$$

$$= G\left(\theta_A\left(\vec{r},\vec{u}\left(t_A\right)\right),\phi_E\left(\vec{r},\vec{u}\left(t_A\right)\right)\right) \quad (2.26)$$

$$= G\left(\vec{r},\vec{u}\left(t_A\right)\right) \quad (2.27)$$

$$= G\left(\vec{s}_R\right) \quad (2.28)$$

Thus any one of the equations 2.25 to 2.29 can be used in the radar echo model. There is no velocity vector modelled which can be used to calculate changes in angle along a curved path. This has been left out as these changes are very small for the trajectories of strip map SAR that try and follow the nominal flight path.

NB this thesis uses monostatic radar therefor:

$$G = G_r = G_t \quad (2.29)$$

where G_t is the gain of the transmitting antenna and G_r is the gain of the receiving antenna. Thus

$$\sqrt{G^2} = \sqrt{G_r G_t} \quad (2.30)$$

$$= \sqrt{GG} \quad (2.31)$$

$$= G \quad (2.32)$$

2.4 Modelling Return Voltage

The radar equation is used to calculate the average received power reflected back from a target to the radar antenna. By using this equation as a guide, a simple model of the recorded voltage strength of the returned signal is derived.

The radar equation given in [13] and [23] represents the average received power reflected from a target at a range R for a particular wavelength of electromagnetic radiation λ given by:

$$P_r = \frac{P_t G_t G_r \sigma \lambda^2}{(4\pi)^3 R^4} \quad (2.33)$$

$$= \frac{P_t G^2 \sigma \lambda^2}{(4\pi)^3 R^4} \quad (2.34)$$

where P_r is the received power, P_t represents the average power transmitted by the radar, $G^2 = G_t G_r$ represents the gain which a signal will undergo, σ is called radar cross section (RCS) and represents the amount of incident power intercepted by the target and re-radiated back towards the radar per unit solid angle by the incident power flux/4π steradian [13]. Using the above equation a model for the average recorded voltage due to the average received power is derived.

The recorded voltage of the signal can be expressed as:

$$V_r \propto \sqrt{P_t} \quad (2.35)$$

$$\propto V_t \sqrt{\frac{G^2 \lambda^2 \sigma}{(4\pi)^3 R^4}} \quad (2.36)$$

$$\propto \frac{V_t G}{4\pi R^2}\sqrt{\frac{\sigma\lambda^2}{4\pi}} \tag{2.37}$$

$$\propto \frac{V_t G}{4\pi R^2}\zeta \tag{2.38}$$

$$= C_K \frac{V_t G}{4\pi R^2}\zeta \tag{2.39}$$

where V_r is proportional to the recorded voltage of the received signal, V_t is the peak transmitted voltage used to transmit a signal, $G = \sqrt{G_t G_r}$ and is the gain of the antenna, R is the range. The radar echo model will thus be a representation of a recorded received voltage which is directly proportional to a true voltage value received.

For simplicity the expression $\sqrt{\frac{\sigma\lambda^2}{4\pi}}$ in equation 2.37 is replaced by the variable ζ thus dropping the radar cross section σ and frequency dependent component λ. The RCS is quite complex to compute as it is dependent on surface area and angle of incidence. This thesis is concerned with focusing. The correct computation for the RCS value is more important to detailed simulation of received SAR data. Thus the expression $\sqrt{\frac{\sigma\lambda^2}{4\pi}}$ is redefined to ζ representing the reflective coefficient of a target which is neither dependent on area angle of incidence or frequency. This has been done so that the modelling of a received signal is much simpler.

2.5 Matched Filtering

Radar operates in a very noisy environment. The radar pulse must be able to compete with noise which may be much stronger than reflected pulses returning from targets to the antenna. A well known process called *matched filtering* is used to increase the signal to noise ratio of a pulse on reception. It is a good technique for the optimal detection of a pulse of known shape which may be delayed in time and scaled in amplitude [12].

The ability to detect returning pulses from targets is, according to matched filter theory [12, 18], dependent on the total energy in the returning pulse. Matched filtering increases the signal to noise ratio by passing the data through a filter - called a *matched filter*. This filter is designed so that the output would greatly increase in amplitude if the correct pulse passes though the matched filter. The filter output responds weakly, in the absence of the correct pulse for which it was designed. The filter will however peak with a voltage amplitude which is proportional to the energy in the pulse if the correct pulse passes through.

The mechanics can be easily explained in the frequency domain. Matched filters are produced specific to the pulse that they are intended to retrieve. The explanation will be restricted to any pulse $p(t) \Leftrightarrow P(f)$ that has a finite bandwidth Δf and has spectral components $|P(f)|$ which can be approximated as:

$$|P(f)| \approx A_{ave} Rect\left(\frac{f}{\Delta f}\right) \tag{2.40}$$

in the frequency domain, where A_{ave} is the average amplitude. The time domain and frequency domain representation of the matched filter for the pulse $p(t)$, with additive noise with a flat (white) power spectral density, is given by:

$$p_{mf}(t) = p^*(-t) \Leftrightarrow P_{mf}(f) = P^*(f) \tag{2.41}$$

where $p_{mf}(t)$ is the impulse response of the matched filter, and $P_{mf}(f)$ is the transfer function. The frequency domain representation (transfer function) of the matched filter is the frequency domain conjugate of the original pulse. This is the key to the detection of the pulse in the time domain. Filtering (time domain convolution) is equivalent to multiplication in the frequency domain. Thus passing the correct pulse through the matched filter would result in a frequency domain product that is totally free of imaginary harmonics due to the conjugate multiplication. The operation of the matched filter is given by:

$$Q(f) = P^*(f) \times (P(f) + n(f)) \tag{2.42}$$
$$Q(f) = P^*(f) \times P(f) + P^*(f) \times n(f) \tag{2.43}$$
$$Q(f) = |P(f)|^2 + P^*(f) \times n(f) \tag{2.44}$$

where $P(f)$ is the spectrum of the pulse $p(t)$ and $n(f)$ is the spectrum of the noise $n(t)$.
The match filter response $q(t) \Leftrightarrow Q(f)$ is dependent on the energy in the pulse $p(t)$. It can be seen that the voltage output $q(0)$ at the centre of convolution between the matched filter $p_{mf}(t)$ and pulse $p(t)$ is related to the energy in the pulse given by:

$$A_{peak} = \int_{\infty}^{-\infty} |P(f)|^2 \, df \tag{2.45}$$

The average noise power output of the matched filter in response to the noise $n(t)$ is given by $\overline{|n_0(t)|^2}$ which is the mean square value of the noise and is independent of time. The requirement for the pulse $p(t)$ to be detected is dependent on the ratio $\frac{A_{peak}^2}{\overline{|n_0(t)|^2}}$. For the special case where the power spectral density is flat, $S_n(f) = \frac{\eta}{2}$, i.e. where η is constant (white noise), the output ratio of matched filter due to the pulse $p(t)$ and the noise $n(t)$ is given in [18] as:

$$\frac{A_{peak}^2}{\overline{|n_0(t)|^2}} = \frac{E}{\frac{\eta}{2}} \tag{2.46}$$

where $E = \int_{\infty}^{-\infty} |P(f)|^2 \, df$. Thus detection of a pulse using a matched filter is only dependent on the energy in the pulse. If the energy in the received pulse $E \geq \frac{\eta}{2}$, then the pulse $p(t)$, at $t = 0$, would in theory cause the output of matched filtered to peak thus allowing the pulse to be detected. The shape of the output for the pulse $p(t)$ is dependent on the shape of the magnitude of the spectral components as approximated in equation 2.40 and is given as:

$$p_o(t) = \mathcal{F}^{-1}\left\{|P(f)|^2\right\} \tag{2.47}$$
$$\approx \mathcal{F}^{-1}\left\{A_{ave}^2 Rect^2\left(\frac{f}{\Delta f}\right)\right\} \tag{2.48}$$
$$= A_{ave}^2 \Delta f \frac{sin(\pi \Delta f t)}{\pi \Delta f t} \tag{2.49}$$
$$= A_{peak} \frac{sin(\pi \Delta f t)}{\pi \Delta f t} \tag{2.50}$$

The complete expression for the output of the matched filter is given by:

$$q(t) = p_o(t) + n(t) \otimes p_{mf}(t) \tag{2.51}$$

and the output at $t = 0$ is given by $q(0) = A_{peak} + n(t) \otimes p_{mf}(t)|_{t=0}$.

The large output of the expression A_{peak} is achieved because all frequency components of the waveform $p(t)$ constructively add in phase at exactly the point in time $t = 0$ [18].

2.6 Radar Echo Modelling

The next three subsections use the theory which has been discussed up till now to build a model for range compressed SAR data. The model is used to build the simulator which produces the range compressed SAR data for inversion by the various focusing algorithms.

2.6.1 Downrange Profile Modelling

For the purpose of showing how the radar echo model is derived this detailed step by step derivation of a single point target is described for a single downrange profile. The transmission pulse used to start the derivation must meet the requirements of those used in real SAR data acquisition. These requirements can be found in [13] section 6.7. The requirements for choosing a pulse which will allow good range compression must allow for:

1. Good target detection which is dependent on the total energy in the illuminating waveform [13].
2. Good range accuracy and resolution which require larger energy and bandwidth in the waveform [13].
3. Good Doppler accuracy and resolution which require pulses which are long in time span and have a reasonable peak power [13].

A linear *frequency modulated* (FM) sweep called a *Chirp Pulse* is the most commonly used pulse that fulfils these requirements. The chirp pulse's bandwidth is built up over time as it is transmitted. Thus the energy that goes into the pulse is gradually spread out over the time it is being transmitted, resulting in a high energy pulse. The basebanded version of the chirp pulse used in this thesis is given by:

$$p(t) = e^{j2\pi\left(\frac{\Delta f}{\Delta t}t^2 - \frac{\Delta f}{2}t\right)} Rect\left(\frac{t}{\Delta t}\right) \tag{2.52}$$

where $\Delta f$is the bandwidth of the transmitted pulse and Δt is the ramp time of the transmitted pulse. The bandwidth of a chirp pulse is approximately the same as the bandwidth over which it ramps [12]. Signal analysis of this fact is not discussed in this thesis but from experimentation and observation it was found to hold true for most cases where the product of $\Delta f \Delta t > 100$ [12, 19]. The product $\Delta f \Delta t$ is known as the *Dispersion Factor* [12].

The modulated chirp pulse that is transmitted, ramps lineally up in frequency over a short period of time Δt from an initial lower frequency $f_i = f_c - \frac{\Delta f}{2}$ around a centre frequency f_c over a bandwidth Δf up to a final high frequency $f_f = f_c + \frac{\Delta f}{2}$. The complete modulated chirp signal as explained above is given by:

$$p_{TX}(t) = V_t p(t) e^{j2\pi f_c t} \quad (2.53)$$

$$= V_t e^{j2\pi\left(\frac{\Delta f}{\Delta t}t^2 + \left(f_c - \frac{\Delta f}{2}t\right)\right)} Rect\left(\frac{t}{\Delta t}\right) \quad (2.54)$$

Where V_t is the transmitted voltage, $p(t)$ is the basebanded chirp pulse and $e^{j2\pi f_c t}$ is the carrier with frequency of f_c.

2.6.1.1 Downrange profile derivation for a single point target

The returned version of the chirp pulse which is reflected off a point target is a delayed, scaled version in time of the original pulse. The time taken for the pulse to travel from antenna to target and back again is directly related to the range from antenna to target, is modelled as:

$$\tau = \frac{2\left|\vec{r} - \vec{u}(t_A)\right|}{c} \quad (2.55)$$

where τ is the time delay of the returning reflected signal. The vector $\vec{u}(t_A)$ is the position of the radar in 3D scene at the start of the pulse transmission and the vector $\vec{r} = \langle x, y, z \rangle$ represents the position of a target in the 3D scene. The received signal still contains the carrier which allows the spectral components to be centred around f_c in the frequency domain. The representation of the returned chirp pulse is given by:

$$p_{RX}(t) = V_r p_{TX}(t - \tau) \quad (2.56)$$

$$= V_r p(t - \tau) e^{j2\pi f_c (t-\tau)} \quad (2.57)$$

where:

$$V_r = \frac{V_t G \zeta}{4\pi R^2} \quad (2.58)$$

$$= \frac{V_t G \zeta}{4\pi \left|\vec{r}_f - \vec{r}_i\right|^2} \quad (2.59)$$

is derived from equation 2.39 with $\vec{r}_f$ representing the target position in the 3D scene and $\vec{r}_i$ representing the radar position in space at the start of the pulse transmission. For simplicity the proportionality constant in equation 2.58 and 2.59 was dropped by setting $C_K = 1$ to avoid carrying unnecessary constants in the the radar echo model.

The received signal is now mixed down to baseband using an I.Q. down converter [12]. The resulting waveform is now expressed as:

$$q_{RX}(t) = V_r p_{RX}(t) e^{-j2\pi f_c t} \quad (2.60)$$

$$= V_r p(t - \tau) e^{-j2\pi f_c \tau} \quad (2.61)$$

In order to distinguish the reflected signals from the surrounding noise, matched filtering must be used to enhance the signal to noise ratio and then recover the position of the resulting reflected chirp pulse. Since the carrier is no longer part of the signal due to the down mixing of the I.Q.

converter, the match filter must now match the basebanded version of the chirp pulse $p(t)$. The representation of the matched filter is given by:

$$p_{MF}(t) = p^*(-t) \Leftrightarrow P_{MF}(f) = P^*(f) \tag{2.62}$$

Using the theory in Section 2.5 and the expression of equation 2.50 the match filtering of the basebanded pulse is given by :

$$q_{MF}(t) \approx V_r A_{peak} \frac{sin(\pi \Delta f(t-\tau))}{\pi \Delta f(t-\tau)} e^{-j2\pi f_c \tau} \tag{2.63}$$

$$\approx \frac{V_t G \zeta A_{peak} \frac{sin(\pi \Delta f(t-\tau))}{\pi \Delta f(t-\tau)} e^{-j2\pi f_c \tau}}{4\pi |\vec{r}_f - \vec{r}_i|^2} \tag{2.64}$$

The above expression was derived by using [12] as a guideline to develop the down-range profile of a single point target return.

Note the following:

- The expression $e^{-j2\pi f_c \tau}$ is a constant and does not take part in the convolution process of matched filtering.
- The Variable V_r has also been replaced by its expansion, namely equation 2.39.

Using the relation that the two-way distance travelled by the pulse is given by $|\vec{r}_f - \vec{r}_i| = \frac{c\tau}{2}$, equation 2.64 above takes on a form only involving time t as an independent variable, as follows:

$$q_{MF}(t) \approx \frac{V_t \zeta G A_{peak} \frac{sin(\pi \Delta f(t-\tau))}{\pi \Delta f(t-\tau)} e^{-j2\pi f_c \tau}}{\pi (c\tau)^2} \tag{2.65}$$

$$\approx \frac{V_t \zeta G A_{peak} Sa(\pi \Delta f(t-\tau)) e^{-j2\pi f_c \tau}}{\pi (c\tau)^2} \tag{2.66}$$

2.6.2 Phase History Modelling

Since the SAR scene is made up of many downrange profiles at different positions along the azimuth the model must now be extended to a 2 dimensional version. The variable t in equation 2.66 is now replaced with t_R as its independent variable, as follows:

$$q_{MF}(t_R) = \frac{V_t G \zeta A_{peak} f Sa(\pi \Delta f(t_R - \tau)) e^{-j2\pi f_c \tau}}{\pi (c\tau)^2} \tag{2.67}$$

- As the radar moves through the scene the distance to the target changes. Thus the time delay τ can be replaced by equation 2.55.
- Since the model is derived for a linear propagation medium the centre frequency f_c and bandwidth Δf of the pulse $p(t_R)$ can be expressed in spatial frequencies where κ_c is the spatial centre frequency and $\Delta\kappa$ is the spatial bandwidth. Since $R = \frac{ct}{2}$ a similar relation can be written for frequencies as $\frac{1}{\kappa} = \frac{c}{f2}$, thus $\kappa = \frac{f2}{c}$. The centre frequency f_c and bandwidth

Δf can now be represented as a spatial centre frequency κ_c and spatial bandwidth $\Delta\kappa$ by substituting the relation:

$$f_c = \frac{c\kappa_c}{2} \tag{2.68}$$

and

$$\Delta f \approx \frac{c\Delta\kappa}{2} \tag{2.69}$$

into equation 2.67.

- the variable ζ has been extended to representing the reflection of a point anywhere in space given by:

$$\zeta(\vec{r}) \tag{2.70}$$

Thus by substituting equations 2.9, 2.25, 2.55, 2.69, 2.68 and 2.70 into equation 2.67 a complete phase history which is related to one point-target with the range locus $R = |\vec{r} - \vec{u}(t_A)|$, is given by:

$$q_{MF}(R, t_A, \vec{r}) = \frac{\zeta(\vec{r})\, A_{peak} V_t G(\vec{r}, \vec{u}(t_A))\, Sa(\pi\Delta\kappa(R - |\vec{r} - \vec{u}(t_A)|))\, e^{-j4\pi\kappa_c \left|\vec{r} - \overrightarrow{u(t_A)}\right|}}{4\pi |\vec{r} - \vec{u}(t_A)|^2} \tag{2.71}$$

The important variables for SAR focusing are $\zeta(\vec{r})$, the phase $e^{-j4\pi\kappa_c|\vec{r}-\vec{u}(t_A)|}$and the locus $R = |\vec{r} - \vec{u}(t_A)|$ representing the peak of the Sa expression of equation 2.71. These all distinguish the phase histories from those related to other point-targets. The phase $e^{-j4\pi\kappa_c|\vec{r}-\vec{u}(t_A)|}$ is the most important factor that uniquely identifies the phase histories and is used for constructive complex multiplication in SAR inversion. The aim of SAR focusing will be to recover the ratio of ζ of one point-target in relation to other point-targets. The complete phase history for a single point-target is given by equation 2.71 and is displayed in Figure 2.6

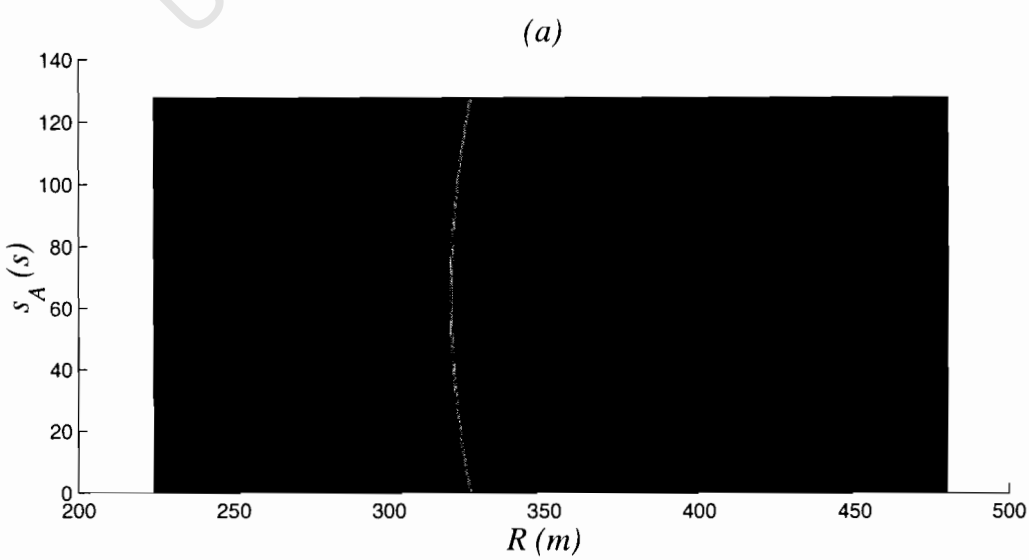


Figure 2.6: Phase history for 1 point-target

2.6.3 Range Compressed SAR Data Model

The resulting equation 2.71 represents a model of a single reflective target in the observed scene. A real SAR image consist of many of these expressions added together using superposition to give the resulting raw SAR data. First equation 2.71 is expanded to:

$$q_{MF}\left(R, t_A, \vec{r}\right) = \zeta\left(\overrightarrow{r}\right) q\left(R, t_A, \vec{r}\right) \tag{2.72}$$

where:

$$q\left(R, t_A, \vec{r}\right) = \frac{q_{MF}\left(R, t_A, \vec{r}\right)}{\zeta\left(\overrightarrow{r}\right)} \tag{2.73}$$

is the the phase history of a ideal point target. The expression for multiple targets are given by equation 2.74:

$$g\left(R, t_A\right) = A \sum_{k=1}^{\infty} q_{MF}\left(R, t_A, \vec{r}\right) \tag{2.74}$$

$$= A \sum_{k=1}^{\infty} \zeta\left(\vec{r}\left[k\right]\right) q\left(R, t_A, \vec{r}\left[k\right]\right) \tag{2.75}$$

$$= A \sum_{k=1}^{\infty} \frac{\zeta\left(\vec{r}\left[k\right]\right) G\left(\vec{r}\left[k\right], \vec{u}\left(t_A\right)\right) Sa\left(\pi \Delta\kappa\left(R - \left|\vec{r}\left[k\right] - \vec{u}\left(t_A\right)\right|\right)\right) e^{-j4\pi\kappa_c\left|\vec{r}[k] - \vec{u}(t_A)\right|}}{4\pi\left|\vec{r}\left[k\right] - \vec{u}\left(t_A\right)\right|^2} \tag{2.76}$$

where: $\vec{r}\left[k\right] = \left\langle x\left[k\right], y\left[k\right], z\left[k\right]\right\rangle$, $A = A_{peak} V_t$ and k is a counting variable. The raw SAR data is a function only dependent on t_A and R. The model is comprehensive enough to simulate the following:

1. A nonlinear aperture e.g. circular, parabolic, or any other arbitrary flight path.
2. The phase history of any point in 3D scene and not only points on a flat surface.
3. The effect of the beam pattern on the phase history of any point-target in a 3D scene.

The model gives no information about the signals, like the pulse length or spectral components, which produced the matched filtered phase history. It only gives information about the centre frequency and bandwidth of the original pulse. In Chapter 5 the SAR simulator used to produce raw SAR data uses this model $g\left(t_A, R\right)$.

Chapter 3

SAR Inversion using Backprojection

Backprojection is an inversion method used in the reconstruction of an image from its internal projection profile. The mathematical mapping which relates a projection profile to an object is called the Radon Transform. Backprojection is thus one of the methods for computing the inverse of the Radon Transform by reconstructing the image from its Radon projection.

The Radon transform is the most widely used tools in medical topographical and topographical image reconstruction. The transform was discovered by J. Radon in the early 1900s and he published a paper in 1917 [15]. An English translation of this paper can be found in appendix A of [15].

The recorded profile of an object can be of many types. It could represent the attenuation of energy or the time delay of a signal either passing through or being reflected off an object. Thus the recorded projection profile of an object can be called the Radon transform of the object [15]. The scene under investigation is called the the observed (object/real) scene while the recorded projection profile is referred to as the projected scene.

Backprojection in SAR differs from that found in medical image processing. SAR is produced by the radar illuminating the scene at different points in space and receiving the reflected echoes. Thus the raw SAR data can be thought of as a "Radon projection" where the phase history represents the flight time of a reflected signal. A brief description of the theory of backprojection as used in medical image processing can be found in Appendix A. This section assumes that the reader is already familiar with the theory of backprojection as discussed in Appendix A.

3.1 SAR Backprojection Theory

For simplicity the "Radon" projection and backprojection equations will be explained in 2-D using Cartesian co-ordinates (x, y), on the $x - y$ plane positioned at $z = 0$ in a 3-D scene. The vector representing a point relative to the origin is given as $\vec{r} = \langle x, y \rangle$. The explanation in this section will also be restricted to linear apertures with flight path given by:

$$\vec{u}(t_A) = \langle u_x(t_A), u_y(t_A) \rangle \tag{3.1}$$

$$= \langle 0, y \rangle \tag{3.2}$$

$$= \langle 0, u \rangle \tag{3.3}$$

where $u = y$ is the nominal flight path axis. The radar used here produces an isotropic beam.

3.1.1 SAR Data Acquisition

Raw radar data are acquired by illuminating a scene under investigation and then recording the received instantaneous signal which is reflected off objects in the scene. This is done by sampling the instantaneous signal received by the antenna over a short period of time producing an array which relates signal received to signal flight time (time delay) called a downrange profile.

When the an isotropic radar illuminates a scene the resulting radiation spreads out in a sphere over time with radar at the centre. The energy that is then reflected off objects is sent back to the radar. In a monostatic SAR case the time taken for the signal to return can be directly related to the distance that the signal travels. Since the power which is received at a certain time is related to a constant distance from the radar, all the signal which is received at a particular time which is recorded in a single range bin, does not come from a single target but must come from a region which is at a radius R from the radar as can be seen in Figure 3.1.

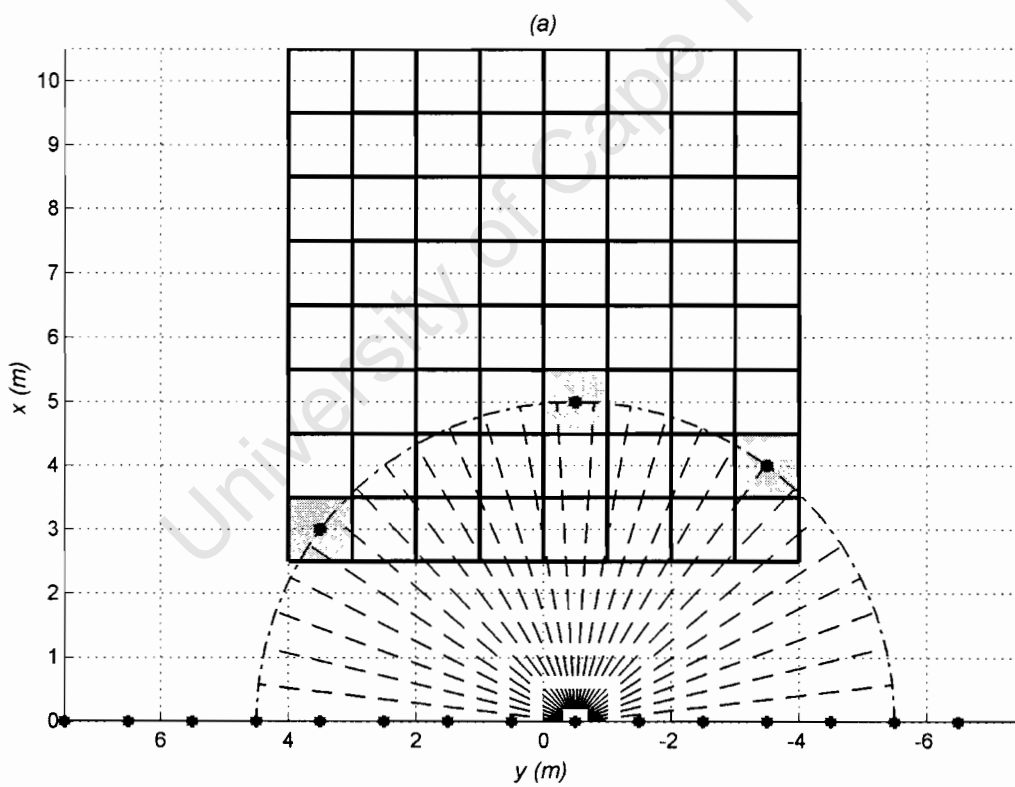


Figure 3.1: 2D scene with three point targets
Targets are positions at $\langle 5, -1\rangle$, $\langle 3, 3\rangle$ and $\langle 4, -4\rangle$ overlaid with image grid. All the signal received at the position $R = 5$ of the downrange profile at position $\vec{u}\,(1) = \langle 0, -1\rangle$ along the flight path comes from a region along the circumference of the semi circle. Thus it can be seen that any possible target on the circumference of the semi circle, either inside or outside of the image grid, would, if it existed, also contribute to the signal received.

NB: In 2D the signal accumulated from point targets all fall on the circumference of a circle instead of the surface of a sphere as in 3D

By repeating the above mentioned process at adjacent positions along the flight path a different downrange profile at each position in the scene is produced due to the change in angle and range from radar to scene targets. The collection of these downrange profiles produces a 2D set of raw SAR data that is used to produce a focused SAR image as shown in Figure 3.2. Due to the change in angle and range from radar to scene targets at each point along the flight path where a downrange profile is recorded, each point target in the scene will produce a hyperbolic loci on the 2D set of raw SAR data which is unique to the target position in the scene. Figure 3.2 shows the recorded information of Figure 3.1 and is the "projection" or inverse of Figure 3.1.

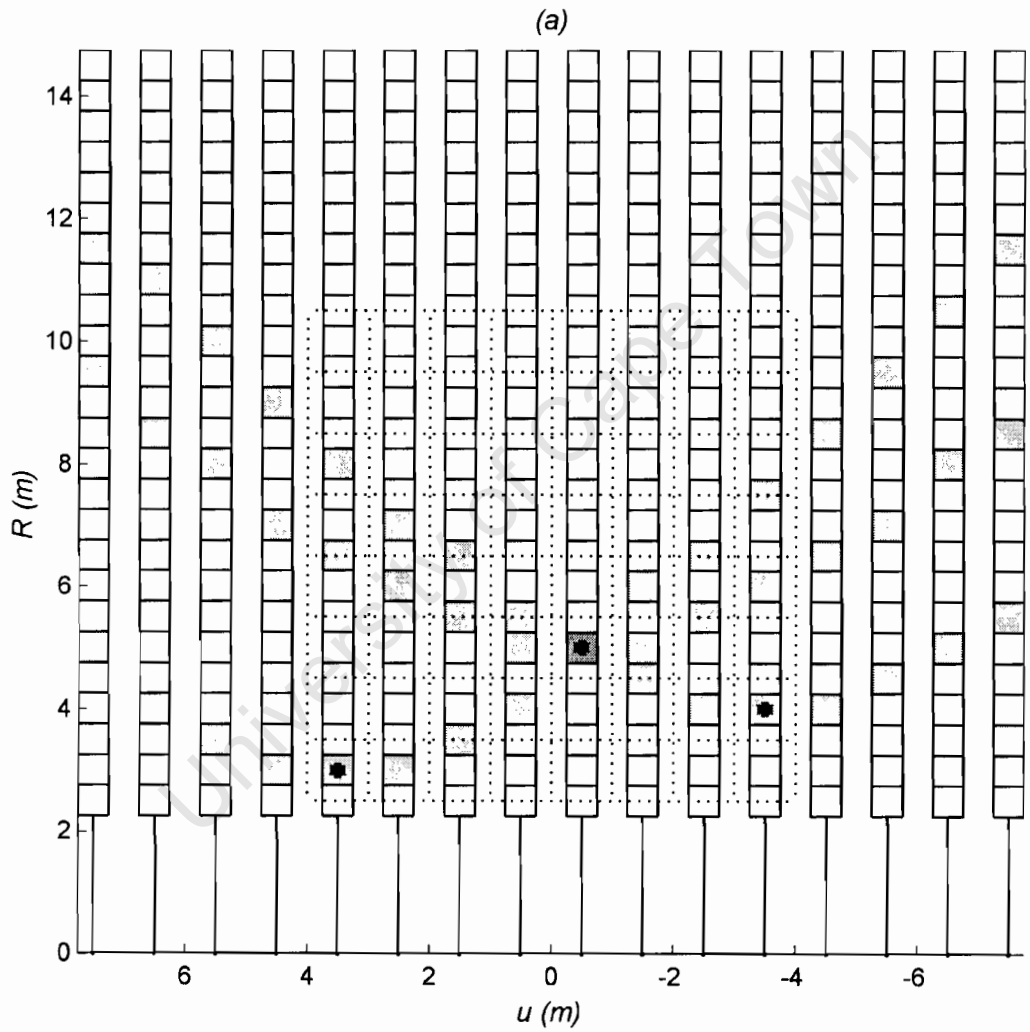


Figure 3.2: Downrange profiles with the received signal.
The diagram shows the 16 downrange profiles with the received signal from the three point targets in Figure 3.1. It can be seen that the Raw SAR data is an accumulation of hyperbolic loci from each point target in the scene.

For the case where the pulse $p = \delta(R)$ and the spacing between sampling points along the flight path and downrange profile approaches the limiting case of zero, the mathematical model for the signal received by one range bin forms a locus which is the circumference of a circle of radius R with the radar at the centre is given by [3]:

$$g(R,u) = \frac{1}{\pi}\int_{Circle}\frac{f(x,y)}{4\pi R^2}d\theta \tag{3.4}$$

$$= \frac{1}{\pi}\int_{-\pi}^{\pi}\frac{f(R\cos(\theta), u+R\sin(\theta))}{4\pi R^2}d\theta \tag{3.5}$$

where $f(x,y) = \zeta(\vec{r})$.

Figure 3.3 shows the observed scene in which a point target at location (x_0, y_0) is represented by the function:

$$f(x,y) = \delta(x-x_0, y-y_0) \tag{3.6}$$

on the $x-y$ plane, while the projected scene on which the SAR data relating to the observed scene is recorded, is represented by the function:

$$g(R,u) = \frac{\delta\left(\sqrt{x_0^2+(y_0-u)^2}, u\right)}{4\pi\left(x_0^2+(y_0-u)^2\right)} \tag{3.7}$$

$$= \frac{\delta\left(R-\sqrt{x_0^2+(y_0-u)^2}\right)}{4\pi R^2} \tag{3.8}$$

[1] on the $u-r$ axes. The flight path on a 2D Cartesian $x-y$ plane representing the real scene is given by $\langle 0, u\rangle$.

It can be seen in Figure 3.3 that all signal at a particular coordinate $\langle R, u\rangle$ in the projected scene is the accumulation of signal found on the circumference of a circle with radius R and centre $\langle 0, u\rangle$ in the real scene [3, 2, 1]. The axis R represents the distance from any point $\langle 0, u\rangle$ to the target [3].

3.1.2 SAR Backprojection Inversion

To use backprojection to reconstruct an image from the real scene, the inverse of the equation 3.4 must be used. The point $\langle R, u\rangle$ in the projected space contains the accumulated information of many points in the real scene. This can clearly be seen in Figure 3.3, where all the points on the hyperbolic loci are actually the accumulation of reflected energy from a circular curve of radius R passing through the point $\langle x, y\rangle$. Thus there are many points in the $R-u$ plane which contain information concerning the point $\langle x, y\rangle$ in the observed scene.

[1] Note the expression, in 3.7 and 3.8 are similar to that in 3.9 and 3.10. Equation 3.8 is not a 1D expression but a curtain Dirac on the locus $R = \sqrt{x_0^2+(y_0-u)^2}$ in 2D. A point Dirac delta in 2D, as in equation 3.11, can be created by multiplying two 1D Dirac deltas, which are curtain Dirac that both fall on the locus of a independent lines in 2D as in equation 3.11.

$$f(x,y) = \delta(x,x) \tag{3.9}$$

$$= \delta(x-y) \tag{3.10}$$

$$h(x,y) = \delta(x,y) \tag{3.11}$$

$$= \delta(x)\,\delta(y) \tag{3.12}$$

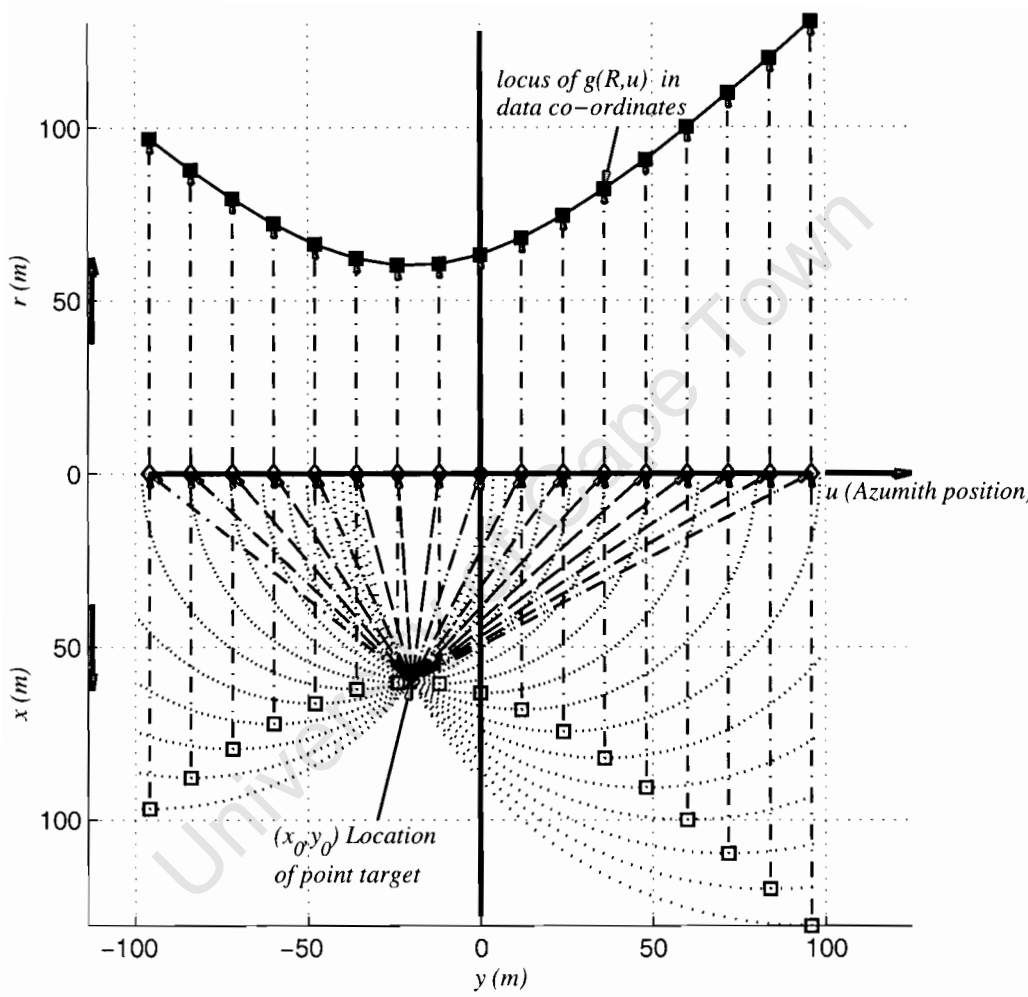


Figure 3.3: SAR Inversion as a transformation

The upper half of Figure 3.3 represents the recorded data in co-ordinates of range R and azimuth u; The lower half on Figure 3.3 shows the scene in (x, y) co-ordinates. The hyperbolic curve $g(R, u)$ is the locus of the projection of a point at position $\langle x_0, y_0 \rangle$ into (R, u) co-ordinates. A particular point $g(R_0, u_0)$ is the superposition of data from circle on the $x - y$ plane with centre $\langle 0, u_0 \rangle$ and radius R.

To find these points a locus of all the points containing information about the point $\langle x, y\rangle$ using the characteristics of SAR data acquisition will be produced. The radar moves along a track $\langle 0, u\rangle$ in the $x-y$ plane. As it moves the distance between a point $\langle x_0, y_0\rangle$ and the point $\langle 0, u\rangle$ changes as u increases as a function of time. The distance is given by :

$$R = |\langle x_0, y_0\rangle - \langle 0, u\rangle| \tag{3.13}$$
$$= \sqrt{x_0^2 + (y_0 - u)^2} \tag{3.14}$$

The variable R represents the distance from the point $\langle 0, u\rangle$ to the point $\langle x_0, y_0\rangle$ which is one point on the circumference of a circle with radius R and centre $\langle 0, u\rangle$. Equation 3.14 represents the locus of all projection points which contain information of the value at point $\langle x_0, y_0\rangle$ in the real scene [3]. Since all points $\langle x_0, y_0\rangle$ are alike, equation 3.13 and 3.14 can be generalised to any point $\langle x, y\rangle$ where:

$$R = |\langle x, y\rangle - \langle 0, u\rangle| \tag{3.15}$$
$$= \sqrt{x^2 + (y - u)^2} \tag{3.16}$$

The backprojection equation can now be written as given in [3] :

$$h(x, y) = \int_{-\infty}^{\infty} g\left(\sqrt{x^2 + (y - u)^2}, u\right) du \tag{3.17}$$
$$= f(x, y) \otimes\otimes \frac{1}{\sqrt{x^2 + y^2}} \tag{3.18}$$

where $\otimes\otimes$represents 2D convolution integral in $x$and y.

Backprojection does not recover the original scene. Instead the resulting output of the backprojection integral contains a $\frac{1}{\sqrt{x^2+y^2}}$ function convolved with the desired output. In order to retrieve the original scene the method of backprojection must be modified [3]. This modification is called Filtered Backprojection and it can be shown that the image can be correctly recovered by passing the projected scene through a ramp filter after backprojection as described in [3] .

$$f(x, y) = \mathcal{F}^{-1}\left(|\kappa_x| \mathcal{F}(h(x, y))\right) \tag{3.19}$$
$$= \mathcal{F}^{-1}\left(|\kappa_x| h(\kappa_x, u)\right) \tag{3.20}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ is the Fourier and Inverse Fourier transforms in the x direction respectively and κ is the spatial frequency in rad/m

Another method of recovering the function $f(x, y)$ is to apply filtered backprojection by changing equation 3.17 to accommodate the ramping while doing the processing in the spatial domain. The addition to the backprojection equation 3.17 is the variable R which is multiplied by the function $g(r, u)$. The output of this inversion fully retrieves the function $f(x, y)$. This type of backprojection is used in [1, 2].

$$f(x,y) = \int_{-\infty}^{\infty} g(R,u)\ R\,du \tag{3.21}$$

where :

$$R = \sqrt{x^2 + (y-u)^2} \tag{3.22}$$

3.1.3 Backprojection Interpretation

The above integral in equation 3.21 can be interpreted in two ways:

1. A pixel in the observed scene can be focused by coherent addition of all points containing information of the pixel in question found on the hyperbolic locus in the projected scene. Thus a focused value $f(x,y)$ at point $\langle x,y\rangle$ of an image can be produced by weighted sum of all points $\langle R,u\rangle$ in the projected space found on the locus:

$$R = \sqrt{x^2 + (y-u)^2} \tag{3.23}$$

2. The focused image can be produced by spreading the information of one bin in the projected scene over a circular locus in the observed scene. Thus the focused image is formed by taking the value at the point $\langle R,u\rangle$ in the projected plane and spreading it out uniformly over the points $\langle x,y\rangle$ which lie on the locus:

$$y = \sqrt{R^2 - x^2} + u \tag{3.24}$$

Figure 3.4 shows the differences between the two backprojection interpretations.

Figure 3.4 (a) shows a point-target at $\langle x_0,y_0\rangle = \langle 4,4\rangle$. The linear distance between the target and each point in the aperture is also shown by the length of the different lines.

Figure 3.4 (b) shows the relative distance to the target at point $\langle x_0,y_0\rangle = \langle 4,4\rangle$ for each point on the aperture of the $r-u$ plane. Here it can be seen that all the points on the hyperbolic loci contain information about the point $\langle x_0,y_0\rangle = \langle 4,4\rangle$.

In Figure 3.4 (d) the point $\langle R_0,u_0\rangle = \langle 4,4\rangle$ contains information about the point $\langle x_0,y_0\rangle = \langle 4,4\rangle$ because it is found on the hyperbolic loci.

However, in Figure 3.4 (c) it is shown that it does not only contain the information of the point $\langle x_0,y_0\rangle = \langle 4,4\rangle$in the $x-y$ plane, but also, contains information of all of the points found on the circumference of the circle centred at $\langle x,y\rangle = \langle 0,u_0\rangle$ on the $x-y$ plane with radius r_0.

- The two methods of backprojection are mathematically identical and produce the same image [1, 2, 3].
- From Sections 3.1.1 and 3.1.2 it can be also be seen that SAR focusing using filtered backprojection is an inversion process [6] which has a one to many mapping. Each point target in the real scene maps to a locus with a hyperbolic nature in the projected scene or each range bin in the projected scene maps to a circular locus in the real scene.

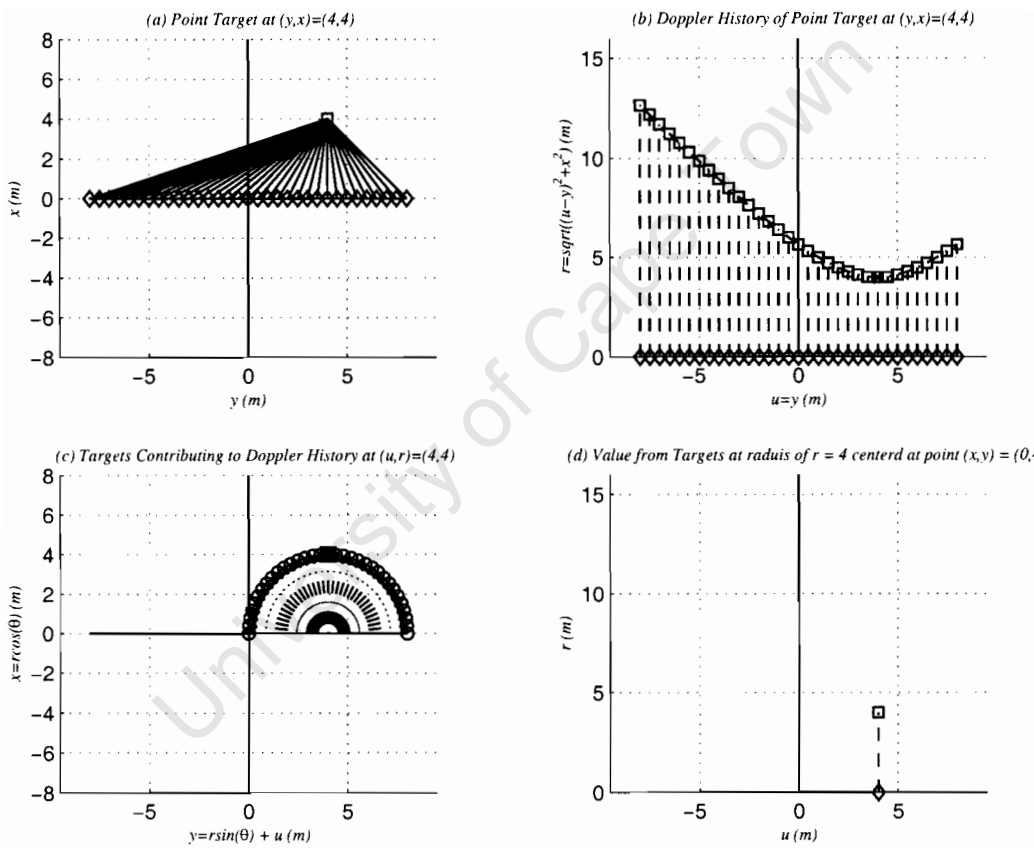


Figure 3.4: Interpretation of backprojection in SAR

- Equation 3.5 is a projection integral which is similar to the Radon transform used in Computer Tomography (CT) [3].
- The inverse of equation 3.5 that focus a SAR scene using backprojection is given by equation 3.17 called Filtered Backprojection. The method termed as an exact inversion method since the reflectivity map of the real scene $f(x, y)$can be completely (exactly) recovered form the projected scene $g(R, u)$.

3.2 SAR Focusing

The whole objective of SAR is to produce 2-D images of the ground surface. To do this the raw SAR data $g(t_A, R)$ must be focused in order to obtain an image. The explanation of SAR is specific to focusing data of the radar echo model produced in Chapter 2 which is unfocused base banded range compressed SAR data with continues variables R and t_A produced by recording the reflected echos of a band limited pulse Δf transmitted into the environment with carrier frequency f_c. This section describes the theory of the backprojection integral used in the thesis project, which produces a band limited representation of the scene based on the theory of exact inversion presented in Section 3.1 .

3.2.1 Backprojection in SAR

Filtered backprojection, as used in the thesis project, is performed on basebanded data which has been matched filtered in range represented by equation 2.74 [14]. Thus unlike Section 3.1 where data is not sampled to baseband the phase component produced by down conversion must be taken into account. Since matched filtering has been performed on the range lines there is only one integral for backprojection of a point in space [14, 11, 1]. Backprojection is given by:

$$h(\overrightarrow{r}) = \int_{t_i}^{t_f} g(R, t_A)\, R e^{j4\pi\kappa_c R} dt_A \tag{3.25}$$

$$= \int_{t_i}^{t_f} g(|\vec{r} - \vec{u}(t_A)|, t_A)\, |\vec{r} - \vec{u}(t_A)|\, e^{j4\pi\kappa_c |\vec{r} - \vec{u}(t_A)|} dt_A \tag{3.26}$$

where: $\kappa_c = \frac{1}{\lambda_c}$ is the spatial carrier frequency, $R = |\vec{r} - \vec{u}(t_A)|$is the range locus along which matched filtering takes place using $e^{j4\pi\kappa_c|\vec{r}-\vec{u}(t_A)|}$ to phase correct the phase history $e^{-j4\pi\kappa_c|\vec{r}-\vec{u}(t_A)|}$ found in the radar echo model to focus a point target at position $\overrightarrow{r}$. The scaling factor $R = |\vec{r} - \vec{u}(t_A)|$ is inserted to compensate for the two way range spreading loss [2]. Equation 3.26 and 3.26 follow the same convention as presented in [1] and [2], for a radar echo model which has a two way range spreading loss that decreases at $\frac{1}{R^2}$.

The focusing integral in equation 3.26 is representative of an isotropic radar. For accurate focusing of a real radar the beam pattern must also be used. Equation 3.27 modifies equation 3.26 by adding the beam pattern $G(\vec{r}, \vec{u}(t_A))$ to the backprojection integral:

$$h(\vec{r}) = \int_{t_i}^{t_f} G(\vec{r}, \vec{u}(t_A))\, g(R, t_A)\; R\, e^{j4\pi\kappa_c R} dt_A \tag{3.27}$$

$$= \int_{t_i}^{t_f} G\left(\vec{r}, \vec{u}\left(t_A\right)\right) g\left(\left|\vec{r} - \vec{u}\left(t_A\right)\right|, t_A\right) \left|\vec{r} - \vec{u}\left(t_A\right)\right| e^{j4\pi\kappa_c\left|\vec{r}-\vec{u}(t_A)\right|} dt_A \tag{3.28}$$

where: $R = \left|\vec{r} - \vec{u}\left(t_A\right)\right|$. To account for the attenuation effect of the beam patten in the raw data, equation 3.28 implements a matched filtering process along the range locus. With reference to equation 2.30, $G\left(\vec{r}, \vec{u}\left(t_A\right)\right) = \sqrt{G_t\left(\vec{r}, \vec{u}\left(t_A\right)\right) G_r\left(\vec{r}, \vec{u}\left(t_A\right)\right)}$.

3.2.2 Ground and Slant Range Focusing

SAR images are either focused to a ground range or in slant range [19]. This thesis project implements all focusing algorithms using ground range focusing; but the theory of both is given in this section.

3.2.2.1 Ground Range Focusing

Using ground range inversion, the focused images can be represented on the $x - y$ plane. This is done by using a height function which models the surface of the ground. The distance z from the $x - y$ plane is given by $z = H\left(x, y\right)$. Each point of the $x - y$ plane which has to be focused uses the function $H\left(x, y\right)$ to retrieve the z distance from the $x - y$ plane. The point in space $\overrightarrow{r}$ is now represented in equation 3.29 by substituting $z = H\left(x, y\right)$, as follows:

$$\overrightarrow{r} = \langle x, y, z \rangle = \langle x, y, H\left(x, y\right) \rangle \tag{3.29}$$

This gives the point on the surface of the height model which must be used for focusing. The backprojection integral which focuses a point to the $x - y$ plane is now given by equation :

$$h\left(x, y\right) = \int_{t_i}^{t_f} G\left(\vec{r}, \vec{u}\left(t_A\right)\right) g\left(R, t_A\right) e^{j4\pi\kappa_c R} R\, dt_A \tag{3.30}$$

where: $\overrightarrow{r} = \langle x, y, H\left(x, y\right) \rangle$ and $R = \left|\vec{r} - \vec{u}\left(t_A\right)\right|$. Figure 3.5 shows a point target focused by equation 3.30.

3.2.2.2 Slant Range Focusing

Instead of focusing an image to the ground range (x, y) plane, slant range inversion focuses to zero-Doppler coordinates (t_s, R_s) where t_s, represents the time in azimuth at which the target makes the smallest angle with the radar boresight during the flight path, and R_s, represents the range at which the target is positioned from the radar.

In this report slant range focusing is approached in a manner similar to ground range focusing. Focusing to ground range coordinates requires the use of a reference point in the scene as in equation 3.29 to focus a point on the (x, y) plane. Using the description of zero-Doppler coordinates in the paragraph above, the reference point in the scene used to focus a point (t_s, R_s) in slant range given by:

$$\vec{r}\left(t_s, R_s\right) = \vec{u}\left(t_s\right) + R_s \left\langle \cos\left(\phi_{Depression}\right) \cos\left(\theta_{Squint}\right), \cos\left(\phi_{Depression}\right) \sin\left(\theta_{Squint}\right), \sin\left(\phi_{Depression}\right) \right\rangle \tag{3.31}$$

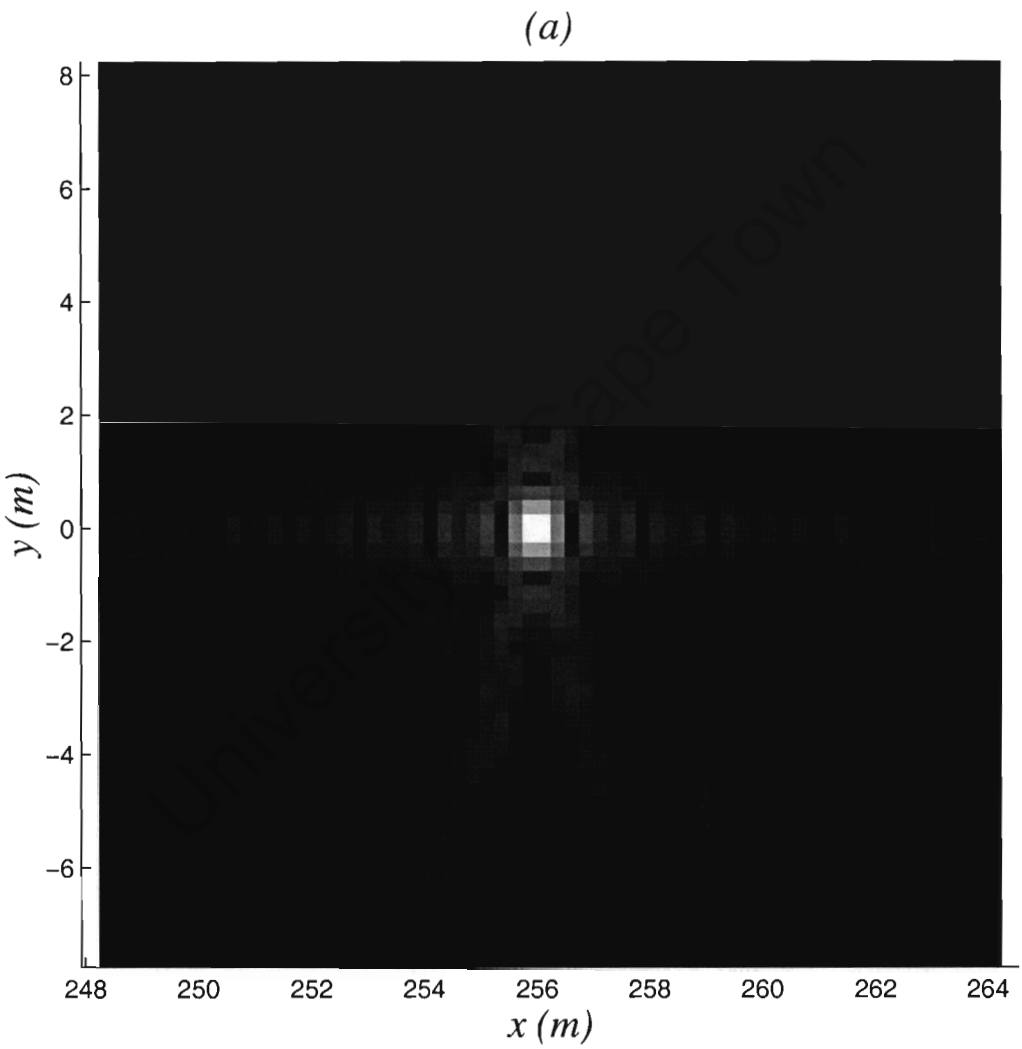


Figure 3.5: Focused point target

Using the vector function $\vec{r}(t_s, R_s)$, a scene is focused in slant range using the backprojection integral is given by:

$$h(t_s, R_s) = \int_{t_i}^{t_f} G(\vec{r}(t_s, R_s), \vec{u}(t_A)) g(t_A, R) e^{j4\pi\kappa_c R} R dt_A \tag{3.32}$$

where: $R = |\vec{r}(t_s, R_s) - \vec{u}(t_A)|$.

For the case where the squint angle $\theta_{Squint} = 0$ and the change in θ_{Squint} angle due to the movement of the radar through space is very small, slant range focusing can be described as "ground range focusing" where the height map $H(x, y)$ is the surface which is traced out by the radar boresight as it moves along the flight path. Figure 3.6 shows the *"slant range surface"* $\vec{r}(t_s, R_s)$ similar to the height function $H(x, y)$ used in ground range focusing.

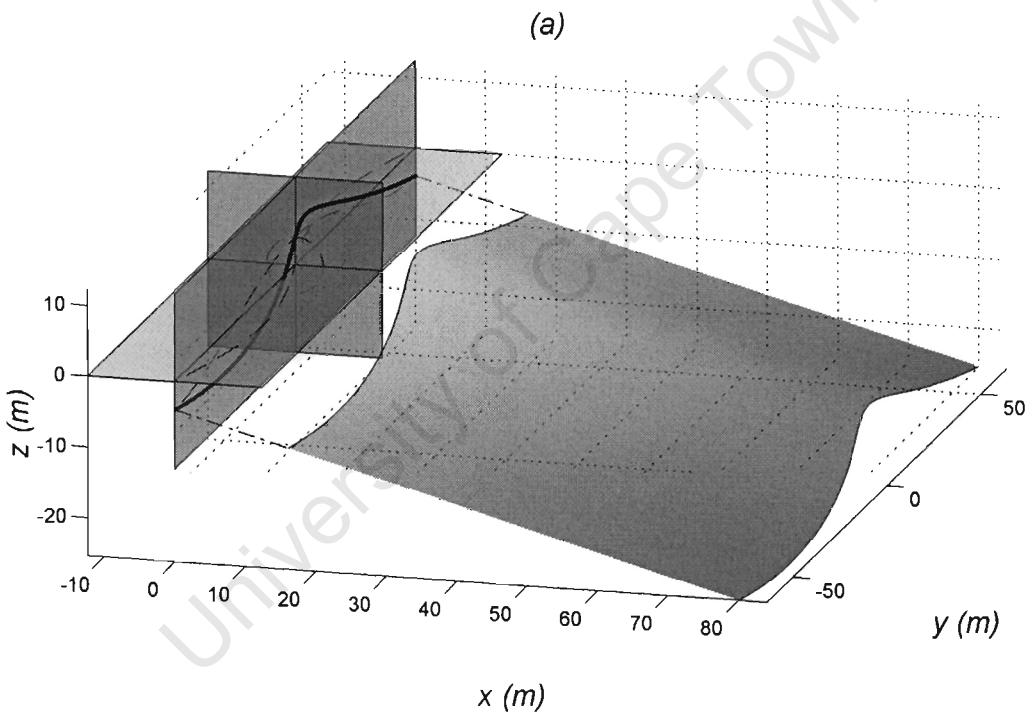


Figure 3.6: Slant range surface

3.3 Frequency Domain SAR Inversion

There is no implementation of frequency domain algorithms done in this thesis. The theory of some frequency domain focusing algorithms which provide exact inversion for a linear aperture can be found in [10], [9] and [5].

The frequency domain analysis of the the LBP and FFBP algorithms are also not described in the thesis. The frequency domain analysis of filtered backprojection using a linear aperture can be found in [1]. A more detailed frequency domain analysis can be found in [7] and [6]. Although there are available frequency domain models which provide exact SAR inversion, they can only be focused to slant range [1]. A brief description of [10], [9] is also given in [1]. These methods of

inversion are similar to filtered backprojection and provide a basis for the filtered backprojection theory used in [1].

3.4 SAR Range & Azimuth Resolution

The theory for the resolution of a focused image is not described within this thesis. Detailed papers on resolution for filtered backprojection used in the thesis project are found in [8] and [7].

For range resolution the thesis project has however used the conventional resolution equations as derived in [12] and [13]. Worked example for resolution can be found in [12]. The resolution in range dependent on the bandwidth Δf of the transmitted pulse. The maximum resolution in range is given by equation3.33:

$$\delta R = \frac{c}{2\Delta f} \tag{3.33}$$

The principal point that makes SAR such an important focusing technique is that lateral resolution is independent of range. It is proved in [13] section 11.7 that the maximum azimuth resolution found in SAR is totally dependent on the antenna diameter given by:

$$\delta L \approx \frac{d}{2} \tag{3.34}$$

where d is the antenna diameter. However this approximation for the azimuth (along track) resolution assumes a small azimuth beamwidth in its derivation and is not valid for wide beam SAR.

A better expression for the resolution in wide beam SAR can be found in [8] and [7] is :

$$\delta A = \frac{\lambda_c}{2\Delta\upsilon}\frac{c}{2\left(f_f - f_i\right)} \tag{3.35}$$

$$= \frac{\lambda_c}{2\Delta\upsilon}\frac{c}{2\Delta f} \tag{3.36}$$

where δA is the ultimate resolving power for ultra wide band and wide beam SAR moving along a linear track [7, 8], $\lambda_c = \frac{c}{f_c}$ is the wavelength of the centre (carrier) frequency, $\Delta\upsilon$ is the beam width and Δf is the transmitted bandwidth $(f_f - f_i)$. The ultimate resolution is obtained when $\Delta f = \frac{c}{\lambda_i}$, $\Delta\upsilon = \pi$ and $\lambda_{c=} = 2\lambda_i$,where $\lambda_i = \frac{c}{f_f}$. The expression is area dependent and it is shown in [8] and [7] that the two expressions $\frac{c}{2(f_f - f_i)}$ and $\frac{\lambda_c}{2\Delta\upsilon}$ both agree with the narrow band case as in 3.33 and 3.34 respectively. Further details of equation 3.35 and 3.36 are discussed in [8] and [7].

Chapter 4

Digital Image Reconstruction

The previous chapter explained backprojection theory for continues signals and variables. The chapter describes the theory of an implementation of the FFBP algorithm carried out at UCT which produces digitally reconstructed ground range images. The algorithm was implemented for the case where the pulse has a finite bandwidth Δf, a flight path which is finite in length, curved and where the data is discretely sampled as is the case for digitally acquired SAR data, similar to that in [1] and [2].

CARABAS[1] used a VHF system making it possible to process data at the first IF stage. Thus papers [1] and [2] describe SAR inversion using the FFBP algorithm for unfocused range compressed SAR data which is still centred around the carrier frequency f_c. The FFBP implementation carried out for this thesis project was done for unfocused range compressed SAR data which is sampled down to baseband.

Firstly one stage backprojection is described which is proved to be proportional to an $\mathcal{O}(n^3)$ process. Then the fast backprojection algorithms used to speed up the computational process of focusing SAR image are explained. The first algorithm is called Fast Factorised Backprojection (FFBP) [1, 2]. The second is called Local Backprojection (LBP) [3].

The FFBP algorithm was the main investigation of the thesis project. It is a multi-stage algorithm for SAR focusing and has a computational saving proportional to an $\mathcal{O}(n^2 \log(n))$ process. Like the FFT, the FFBP uses a divide and conquer approach to reduce the amount of instructions needed to complete the processing task. FFBP uses a constant factor F to divide the main aperture and main images through the S stages of processing. Theoretically using a factor of $F = e$ gives the biggest reduction in instruction count which requires the most number of stages to reconstruct an image. However the quality of the image degrades as more stages are used. The one stage version of the FFBP algorithm is the normal filtered backprojection algorithm [1, 2] . In this case the factor F is set equal to the number of points in the main aperture.

Although the FFBP algorithm is the main objective of this thesis it is stated in [2] that LBP is the FFBP algorithm using two stages. The LBP has some useful practical advantages which allows this algorithm to consume a lot less computational resources during processing.

[1] Coherent All RAdio BAnd Sending (CARABAS) is a SAR research group based in Sweden. The group produced a wide beam synthetic aperture system with high relative bandwidth. CARABAS operates in the VHF band with frequencies ranging from 20 to 90 MHz which enables the radar to penetrate foliage [3].

4.1 One Stage Filtered Backprojection

For the case where data is discrete and sampled down to baseband the backprojection process becomes an approximation of equation 3.30. Instead of having a continuous flight path of infinite length, there is now a discrete flight path of L points given by :

$$\overrightarrow{u}\,[l] = \overrightarrow{u}\,(ldt_A) \tag{4.1}$$

where l is the indexing variable of the discretely sampled flight path with azimuth spacing in time dt_A. The ground image to which data will be focused is now also finite and discretely sampled in both $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ direction. Thus the image $f\,[m,n] \approx f\,(x\,[m]\,, y\,[n])$ is only a band limited approximation of the real scene where:

$$x_m\,[m] = \left(m - \frac{M+1}{2}\right) dx + X_c \tag{4.2}$$

and

$$y_n\,[n] = \left(n - \frac{N+1}{2}\right) dy + Y_c \tag{4.3}$$

where the indexing variables with pixel spacing (dx, dy) of the $[M,N]^2$ pixels in the respective $\langle \hat{\mathbf{x}}, \hat{\mathbf{y}} \rangle$ directions and $\langle X_c, Y_c \rangle$ is the image grid centre. The filtered backprojection process which gives a digitally reconstructed image and is an approximation of equation 3.30 is given by:

$$f\,[m,n] \approx \sum_{l=1}^{L} G\,(\overrightarrow{r}\,[m,n]\,, \overrightarrow{u}\,[l])\, g\,[o_R, l]\, R\,[m,n,l]\, e^{j4\pi\kappa_c R[m,n,l]} \tag{4.4}$$

where

$$R\,[m,n,l] = |\overrightarrow{r}\,[m,n] - \overrightarrow{u}\,[l]| \tag{4.5}$$

with range bin spacing dR and

$$\overrightarrow{r}\,[m,n] = \langle x_m\,[m]\,, y_n\,[n]\,, H\,[x_m\,[m]\,, y_n\,[n]] \rangle \tag{4.6}$$

is the position in the scene used to calculate the range locus for focusing. The range indexing o_R is the given by

$$o_R = \left\lfloor \frac{R\,[m,n,l] - R_I}{dR} + \frac{1}{2} \right\rfloor \tag{4.7}$$

where R_I is the position of the first range bin in the 2D array of unfocused SAR data.

This integral has been dropped and replaced by a simple summation. A more accurate approach would be to implement the integration using the mid point or trapezoid integration method as in [11]. However it was found that a simple summation is still sufficient for accurate focusing.

The phase correcting argument ($e^{j4\pi\kappa_c R[m,n,l]}$) is inserted in equation 4.4 to take into account the effect of the phase factor ($e^{-j4\pi\kappa_c R[m,n,l]}$) which is introduced into the data by down conversion of the signal in the spectral domain to baseband.

[2]The integer values designating image size M and N are not related. The value of M is independent of the value of N.

A graphical interpretation of the summation can be seen in Figure 4.1. To focus a single pixel, all the downrange profiles, of Figure 3.2, are rotated so that they pass through the pixel centre. Thus the range bin centres, which contain information of that pixel will be inside the pixel borders. These range bin values are then weighted, summed and placed in the pixel in question to get a focused pixel value. Computationally, a pixel is thus assigned a value by interpolating the pulse echo data at the time delay corresponding to the range between pixel and antenna [1].

The main drawback is the time consuming nature of this process. In one stage SAR focusing a $L \times O$ block of raw SAR data is focused to a $M \times N$ image. Equation 4.4 shows that each pixel is focused by L range bin values. Inversion of a SAR scene containing L aperture points which must be focused to an $M \times N$ pixel image requires an operation count of $M \times N \times L$ which is proportional to an $\mathcal{O}\left(n^3\right)$ process [11].

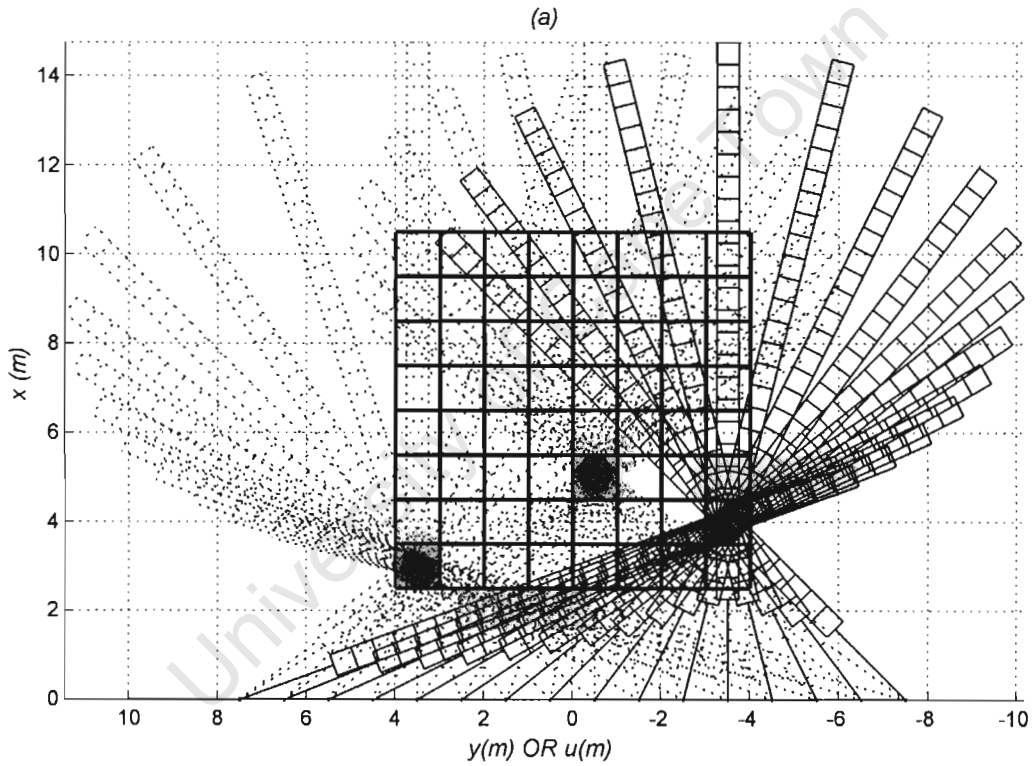


Figure 4.1: Graphical representation of the focusing
The diagram represents the focusing of 3 of the 8×8 pixels on the image grid as performed by equation 4.4.

The computational complexity to focus the image would require LMN looped steps. This is inefficient when focusing SAR data of normal size. A typical image may contain 4096 aperture points and may have to be focused to a grid which is 2048×2048 large. The total number of operations in this case will be $4096 \times 2048 \times 2048 = 17.18 \times 10^9$.

If there is a calculation method which reduces the complexity of the focusing problem and reduces the amount of instructions which is needed to focus an image by splitting up a the single focusing problem, into many smaller problems then this would greatly help to overcome the main drawback of the FBP algorithm.

4.2 Multi Stage Fast Factorised Filtered Backprojection

FFBP is a recursive, multi-stage SAR inversion algorithm which has a computational operation count proportional to $F \log_F (L) NM$ as discussed in [1, 2]. This is when focusing a $M \times N$ pixel image, from L aperture positions and corresponding downrange profiles, through S $(\log_F (L))$ stages of processing, using a factor F which controls the focusing, that reduces the instruction count and in turn speeding up image reconstruction. In comparison to FBP with an operational count proportional to LMN, the FFBP can have up to a maximum saving of $\frac{L}{F \log_F L}$ times. This is if the aperture points L, factor F and stages S are related by:

$$L = F^S = \prod_{s=1}^{S} F \tag{4.8}$$

where s is the indexing variable for the current stage. This means that the number of stages $S = \log_F L$ is directly linked to the factor used to control the focusing process. (Thus for a 2048×2048 pixel image focused by 4096 downrange profiles using a factor of $F = 2$ and stages $S = 12$, the algorithm in theory will processed the scene $170\frac{2}{3}$ times faster than if FBP was used.)

4.2.1 FFBP Data Flow

Before formally describing the FFBP algorithm, an example is given which shows the difference in the way FBP and FFBP focus a pixel. The example is theoretical and takes place on a 2D scene. At this point many signal processing concepts relevant to focusing are not taken into account such as phase correction , the resolution of downrange profile or image and length of aperture in comparison to image size. The flow of information takes precedence to give the reader a visual understanding of how data is processed. The explanation given here is general to FFBP where $L = F^S$, however, the description of events is specific for unfocused SAR data containing $L = 8$ aperture sampling points and corresponding downrange profiles, which is processed to focus a $M \times N = 8 \times 8$ pixel image using a factor $F = 2$, $S = 3$ stages. The explanation will be kept as general as possible.

The understanding of FFBP in contrast to FBP can be explained by using the diagrams shown in Figure 4.2.

4.2.1.1 Figure 4.2 (a)

Figure 4.2 (a) shows where the information needed to focus the pixel in Figure 4.2 (a) (ii) can be found on the downrange profiles. In Figure 4.2 (a) a single focused pixel is produced using equation 4.4. It can be seen that each pixel requires $L(= 8)$ range bin values to focus a pixel. Since there are $M \times N(= 8 \times 8)$ pixels this would require $L \times M \times N$ $(8 \times 8 \times 8 = 512)$ operations.

4.2.1.2 Figure 4.2 (b)

Figure 4.2 (b) shows where the information needed to focus the pixel in Figure 4.2 (b) (iv) can be found on the downrange profiles.

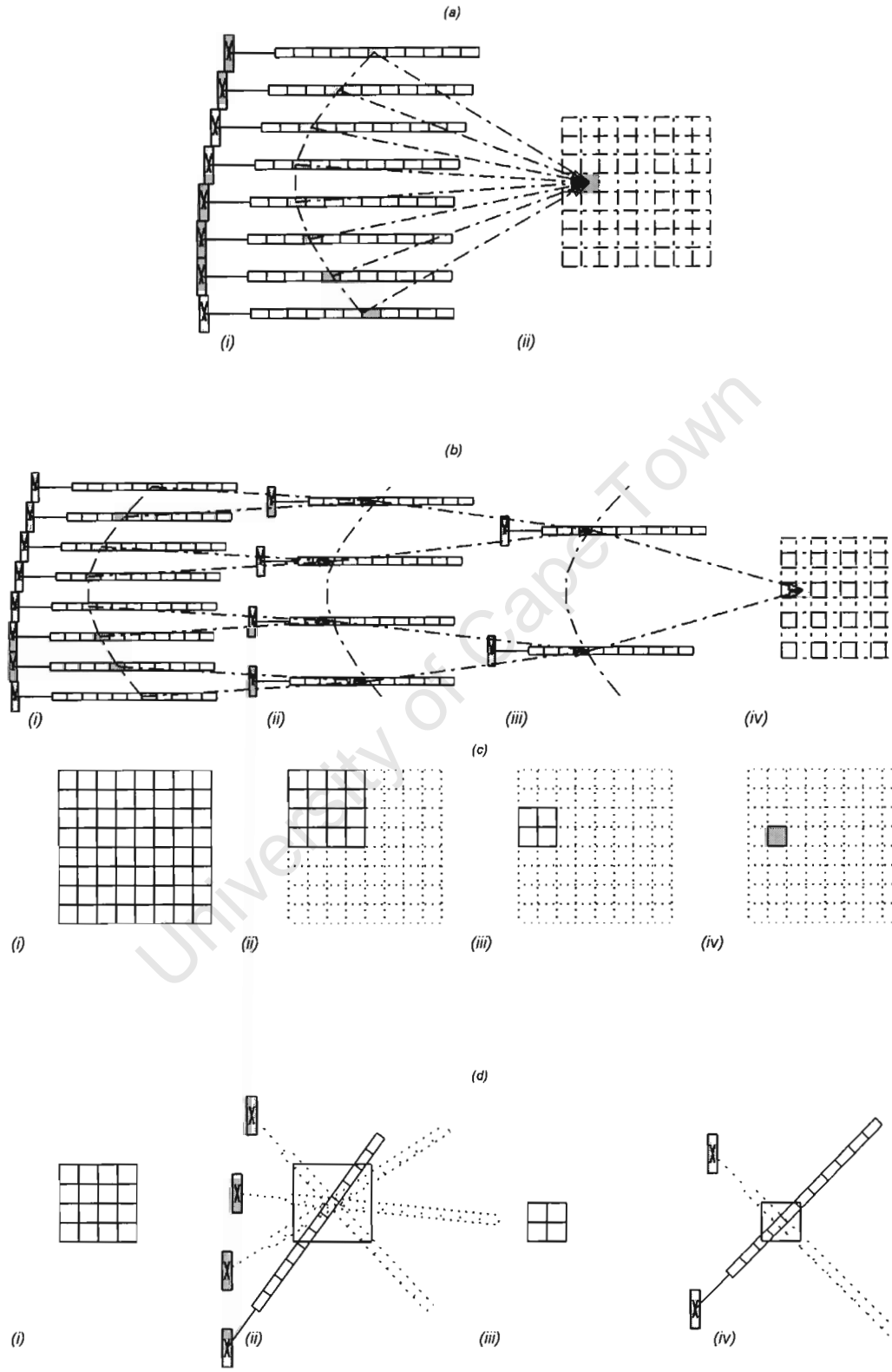


Figure 4.2: Data flow of information to focus a single pixel using FFBP
Focusing of a single pixel using FBP (a) and FFBP (b) for , $M \times N = 8 \times 8$, $S = 3$ and $F = 2$. (1). Image splitting (c) and downrange profile accumulation (d)

The manner in which information is added to the new downrange profile is explained with the aid of Figure 4.2 (d). For now all that must be known is that, instead of solving the problem directly, the focusing problem is done in $S(=4)$ stages. Each stage of processing produces a new number of downrange profiles and corresponding aperture sampling points decrease by a factor $F\ (=2)$. This is done by splitting each number of adjacent downrange profiles and corresponding aperture sampling points into groups $F(=2)$ downrange profiles and corresponding aperture sampling points. Each of these groups will be used to produce a new downrange profile and corresponding aperture sampling point.

- The new aperture sampling point positions are found by finding the average of the $F(=2)$ parent aperture sampling points.
- The new range bin value is found by weighted summing along range loci of all downrange profiles (Filtered backprojection) in the the parent group, and the result is put into the range bin of the new (child) downrange profile on the same range loci .
- The only exception is the final stage where the focused final value is placed in the correct pixel.
- It can be seen that the new set of downrange profiles and corresponding aperture sampling points in Figure 4.2 (b) (ii) & (iii) are both reduced by a factor of $F(=2)$, but approximates the pulse echo data as if it was acquired along the new main apertures.

Thus Figure 4.2 (b) (i) corresponds to the "end" of stage $s = 0$, which is the original data as captured (produced) by the radar. This data forms the start of stage $s = 1$. Figure 4.2 (b) (ii) corresponds to end of stage $s = 1$, which was produced by the data in Figure 4.2 (b) (i). Again, Figure 4.2 (b) (iii) shows the end of stage $s = 2$, which was produced by the data in Figure 4.2 (b) (ii). Finally Figure 4.2 (b) (iv) shows the end of stage $s = 3$, which was produced by the data in Figure 4.2 (b) (iii).

Although a pixel is focused by only F downrange profiles in stage S, it can be seen in Figure 4.2 (b) that the information still comes from L downrange profiles along the range locus. This shows that the information used to focus a pixel in 4.2 (a) is the same information used to focus a pixel in 4.2 (b) . Thus making the two focused values identical in theory.

Specifically for this example, it means that stage $s = \{1, 2, 3\}$ starts out with $L_K = \{8, 4, 2\}$ downrange profiles and corresponding aperture points which will produce $K_L = \{4, 2, 1\}$ downrange profiles and corresponding aperture points by the interpolation of $L_F = \{2, 2, 2\}$ aperture points in each sub-aperture in each stage.

4.2.1.3 Figure 4.2 (c)

Although the data found in the downrange profiles in 4.2 (b) (i) contain all the information to focus the whole $M \times N(= 8 \times 8)$ main image, the downrange profiles in 4.2 (b) (ii) and (iii) have been reduced by a factor F thus reducing the amount of information contained within them. Thus they cannot focus the whole $M \times N(= 8 \times 8)$ main image. Instead, the data in 4.2 (b) (ii) and (iii) can only focus the sub-images as shown in Figure 4.2 (c) (ii) and (iii) respectively.

The pixel in Figure 4.2 (b) (iv) is one of the four pixels in Figure 4.2 (c) (iii) which are focused by data found in the downrange profiles of Figure 4.2 (b) (iii). Thus the downrange profiles in 4.2 (b)

(iii) are not only capable of focusing a single pixel but all four pixels($M_F \times N_F = 2 \times 2 = 4$) in Figure 4.2 (c) (iii). Likewise, the information in the downrange profiles of Figure 4.2 (b) (ii) can be used to focus the sixteen pixel ($M_F \times N_F = 4 \times 4 = 16$) sub-image in 4.2 (c) (ii), without the necessity of processing more stages. Instead, the 4.2 (b) (ii) is processed to produce four more sets of downrange profiles. Figure 4.2 (b) (ii) is one of the four sets.

The manner in which downrange profiles are populated ensure that 4.2 (b) (ii) and (iii) contain all the information to focus the sub-images in Figure 4.2 (c) (ii) and (iii) using FBP. Thus, each new set of downrange profiles will only be able to focus a smaller sub-image than in the previous stage. The sub-images are produced by splitting each sub-image in each stage into $I_M \times J_M = F \times F$ $(= 2 \times 2)$ new sub-images.

In total stage $s = 1$ produces $I_F \times J_F = 2 \times 2$ new sub-images each having a set of $L_K = 4$ downrange profiles. Stage $s = 2$ produces $I_F \times J_F = 4 \times 4$ new sub-images, each having a set $L_K = 2$ downrange profiles. Thus for this example, it means that in each stage, $s = \{0, 1, 2, 3\}$ starts with $I_F \times J_F = \{1 \times 1, 2 \times 2, 4 \times 4, 8 \times 8\}$ sub-images each containing $M_F \times N_F = \{8 \times 8, 4 \times 4, 2 \times 2, 1 \times 1\}$ pixels. This is achieved by splitting each sub-image being locally partitioned into $I_M \times J_N = \{2 \times 2, 2 \times 2, 2 \times 2, 2 \times 2\}$ sub-images.

- The FFBP algorithm requires far fewer instructions, due to the manner in which information to a specific pixel is extracted. Information shared between pixels are not discarded as in FBP, but stored so that it can be used multiple times to focus other pixels. This can be seen in Figures 4.2 (b) and (c) (ii) and (iii). The information which is used to produce the downrange profiles in Figures 4.2 (b) (iii), is the same downrange profiles which will produce three other sets of downrange profiles altogether producing $I_M \times J_N (= 2 \times 2)$ new sets of downrange profiles.

- In contrast, the FBP always focuses a pixel using $L (= 8)$ downrange profiles. Thus information, which could be stored, is discarded after the focusing of a single pixel. The summation needed to focus a single pixel requires unnecessary calculation since common information to focusing adjacent pixels could have been used if stored.

- In short the FFBP which does not focus a single pixel but extracts information common to a collection of pixels and stores this information in each stage. As each stage of the FFBP algorithm progresses, the focusing information becomes more specific to a smaller collection of pixels, up until the collection of pixels is focused by being one pixel in size.

4.2.1.4 Figure 4.2 (d)

Figure 4.2 (d) describes how the information is accumulated, so that the data flow path to focus a single pixel in Figure 4.2 (b) is valid.

The manner in which information is added to the new downrange profile is shown in Figure 4.2 (d).

Figure 4.2 (b) does not show how information is processed, but only where the information needed to focus the pixel is found in Figure 4.2 (b) (iv). The range loci are specific to the pixel being focused and are not the loci on which backprojection takes place. The manner in which range bin values are populated must be done in a general way, so that the information specific to a pixel can

be found on the range loci as shown in Figure 4.2 (b) (i), (ii) and (iii). The manner in which the downrange profiles are populated must ensure that each new L_K downrange profile still contains all the information to focus there corresponding sub-images.

To calculate the range loci over which backprojection must take place, FBP, uses the centre point of each sub-pixel. FFBP ensures that all information in a sub-image is captured, by each one of the L_K downrange profiles being laid over the grid as in 4.2 (d) (ii), so that the downrange profile passes through the sub-image centre. Then, the range bin centres are used to calculate the range loci. Only range bins with range bin centres within the sub-image borders will be populated. As explained above, the new range bin values are found by filtered backprojection using the parent group of $L_F = F$ downrange profiles corresponding aperture sampling points. The result is put into the range bin of the new (child) downrange profile on the same range loci .

It can be seen the the number of downrange profiles, which cross an image, is dependent on the image size. If the image decreases by $I_M \times J_N (= 2 \times 2)$, then the number of downrange profiles, which need to be populated, also decreases by $I_M \times J_N (= 2 \times 2)$. In FFBP ([1], [2] and) in this implementation the number of downrange profiles $O_{MN} = M_F$. In stage $s = 3$ the image pixel is only $M_F \times N_F (= 1 \times 1)$ large. Stage $s = \{1, 2, 3\}$ processes $O_{MN} = \{4, 2, 1\}$range bins.

4.2.2 FFBP Analysis

The FFBP algorithm derives its name from the factor F which is used to factorise the main aperture of L points and partition the main image of $M \times N$ pixels through the S stages of processing. The digital reconstruction of SAR images using FFBP is not as straight forward as FBP. Conceptually there are separate processors which takes place during FFBP image reconstruction. The understanding of these processors gives an understanding of the algorithm.

The previous section looked at the manner in which information flows to produce focus a pixel. Here FFBP is described in a general manner and at a much more technical level. The section also describes the signal processing aspect of focusing. FFBP is explained here for a $L \times O$ range compressed unfocused SAR data set which is focused to a $M \times N$ pixel ground range image using factor F and S stages of processing which are related to the number of L aperture sampling points along the flight path by $L = F^S$.

Sections 4.2.3 and 4.4 derive formulas which allow for the analysis of the instruction count to show that the algorithm indeed performs fewer operations. In addition to the above Sections 4.2.5 and 4.2.6 also give detailed formulas for signal processing. Then Section 4.2.8 and 4.2.9 present different metrics for instruction count and time saving of the algorithm. These sections provide extremely detailed formula for implementation of the algorithm, especially Sections 4.2.5 and 4.2.6.

In this explanation F,L,M,N and S are all positive non zero integers. However due to M or N not always being a factor of F the sub-images and amount of pixels or points may contain fractional parts which in practise are impossible to focus. Variables such as the aperture sampling points and corresponding downrange profiles at the start of a stage ($L_K[s]$), aperture sampling points and corresponding downrange profiles at the end of a stage ($K_L[s]$), number of sample point within a sub-aperture($L_F[s]$), the number of sub-images points in each stage($I_F[s] \times J_F[s]$), the number of pixels in each sub-images ($M_F[s] \times N_F[s]$) and number of sub-images each image will produce on a local level ($I_M[s] \times J_M[s]$) may be affected in this way. Thus issues however will be dealt

with in Chapter 5 where the restriction placed on F and S will be relaxed so that $L \neq F^S$ and constraints placed on this implementation to ensure integer values for the variables above.

4.2.3 Aperture Factorisation

Aperture factorisation is the reduction of aperture sampling points through the S stages of processing.

- Stage $s = 1$ starts off with a main aperture of L aperture points.
- In each stage, a new aperture is produced which is reduced by F aperture sampling points.
- The reduction of aperture points is done by splitting the main aperture in each stage up into sub-apertures each containing F adjacent aperture points. Each of these sub-apertures of F sampling points are interpolated to a new single sampling point.
- A new aperture of K_L points is produced at the end of each stage.
- Thus, stage $s = S - 1$ starts off with F aperture sampling points, which aids in the focusing of information from downrange profiles to an image grid.
- The initial aperture in stage $s = 1$ starts of with $L_K[1] = L$ aperture points.

The explanation of this process can be found in Figure 4.3. A formal description for processing is given below.

In general, each stage will start off with

$$L_K[s] = \frac{L}{F^{s-1}} \tag{4.9}$$

aperture sample points and each stage ends with

$$K_L[s] = \frac{L_K[s]}{F} \tag{4.10}$$

sample points in the aperture.

In each stage, the aperture consisting of $L_K[s]$ sample points is broken up into $K_L[s]$ sub-apertures consisting of

$$L_F[s] = F \tag{4.11}$$

sampling points. Using interpolation schemes like averaging, cubic or spline interpolation [11] the new aperture sample points are found. The initial main aperture $\overrightarrow{u_s}[0, l] = \overrightarrow{u}(ldt_A)$ represents the aperture sampling points of the flight path (as produced by the end of stage $s = 0$). The new apertures for each stage, using averaging as an interpolation scheme are then given by :

$$\overrightarrow{u_s}[s+1, k_l] = \sum_{l_k=1}^{F} \frac{u_s[s, k_l F + l_k]}{F} \tag{4.12}$$

$$= \sum_{l_k=1}^{F} \frac{\overrightarrow{s_u}[s, k_l, l_k]}{F} \tag{4.13}$$

where

$$\overrightarrow{s_u}[s, k_l, l_k] = \overrightarrow{u_s}[s, k_l F + l_k]$$

is the current sub-aperture, k_l is the variable indexing the current sub-aperture and l_k is the indexing variable for an aperture point in the current sub-aperture.

It can be noted, that the number of aperture points at the end of one stage is equal to the number of aperture points in the next stage given by :

$$K_L[s] = L_K[s+1]$$

Also, each stage starts off with

$$L_K[s] = K_L[s] \times L_F[s] \tag{4.14}$$

Since each of $K_L[s]$ sub-aperture sample points produces a new aperture sample point, the number of sub-apertures must equal the number of new sampling aperture points.

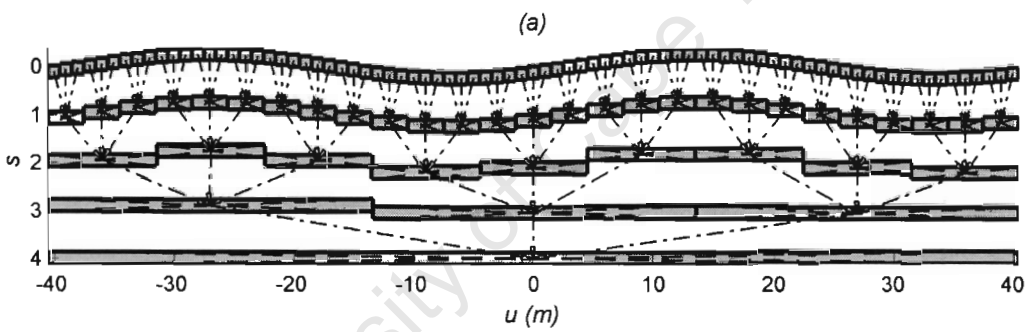


Figure 4.3: Graphical representation of the aperture factorisation

The process is done here for an aperture containing $L = 81$ initial sample points ,using a factor $F = 3$ to control the factorisation process through $S = 4$ stages of processing.

4.2.4 Image Partitioning

In image partitioning, the main image is progressively split up through the S stages of processing.

- Stage $s = 1$starts of with a single main image containing $M \times N$ pixels.
- In stage $s = 1$ the main image is broken up so that there are $F^{S-1} \times F^{S-1}$ pixels in all first stage sub-images.
- In each other stage each sub-images is split up into $F \times F$ sub-images. The size of the sub-images get reduced by $F \times F$ pixels in each stage but the number of sub-images increase exponentially by $F \times F$ images altogether.
- The final stage $s = S$ starts out with $\frac{M}{F} \times \frac{N}{F}$ sub-images which are $F \times F$ pixels in size.
- In the final stage $s = S$ there are $L_K[S] = F$ sample points in the aperture. Since an aperture sample point always correspond to a particular downrange profile, there are F downrange profiles that must focus the $F \times F$ sub-images.

- In theory the $F \times F$ pixels in the sub-aperture are split up into $F \times F$ sub-images each 1×1 pixels in size. Each of these 1×1 pixel images are focused by the F downrange profiles and corresponding aperture sample points. Altogether, there are $M \times N$ of these 1×1 pixel images which must be focused.

Thus, in image partitioning, the main image is progressively split up through the S stages of processing from a single image containing $M \times N$ pixels into $M \times N$ images containing 1×1 pixels each. The main requirement for image splitting is that the image grid in the final stage must be 1×1 pixels in size. A formal description of the process in given below.

The number of images in each stage $(1 \leq s \leq S)$ is given by:

$$I_F[s] \times J_F[s] = \frac{M}{F^S} F^s \times \frac{N}{F^S} F^s \tag{4.15}$$

Thus, for stage $s = S$ there are altogether $I_F[S] \times J_F[S] = M \times N$ images of 1×1 pixels in size.

The pixel size of a sub-image for each stage $(1 \leq s \leq S)$ is given by:

$$M_F[s] \times N_F[s] = F^{S-s} \times F^{S-s} \tag{4.16}$$

From equation 4.15 it can be seen that the number of images increases exponentially as the algorithm progresses through its S stages. The sub-image splitting in FFBP branches out recursively producing a 2D tree. The pixel lines drawn in bold in Figure 4.4 shows an example of one branch of the image partitioning process.

The cumulative number of images in each stage given by equation 4.15 is due to each sub-image being split locally into

$$I_M[s] \times J_N[s] = F \times F \tag{4.17}$$

sub-images for stages $(1 < s \leq S)$.

For stage $(s = 1)$, each image will have $M_F[1] \times N_F[1] = F^{S-1} \times F^{S-1}$ pixels, and therefore the number of sub-images produced by the end of stage $s = 1$ is also equal to the total number of images in those stages:

$$I_M[1] \times J_N[1] \quad = \quad I_F[1] \times J_F[1] \tag{4.18}$$

4.2.5 Downrange Profile Accumulation

The manner in which a downrange profile's range bins are populated is explained below. The process of downrange profile accumulation is general for all range bins. The explanation given here is for the population of one range bin of one downrange profile which is accumulated from F downrange profiles all belonging to a single sub-aperture.

4.2.5.1 Range Bin Population

Each new sub-image receives an interpolated set of downrange profiles which are reduced by a factor F but approximate the representation of pulse echo data as if it was acquired along the

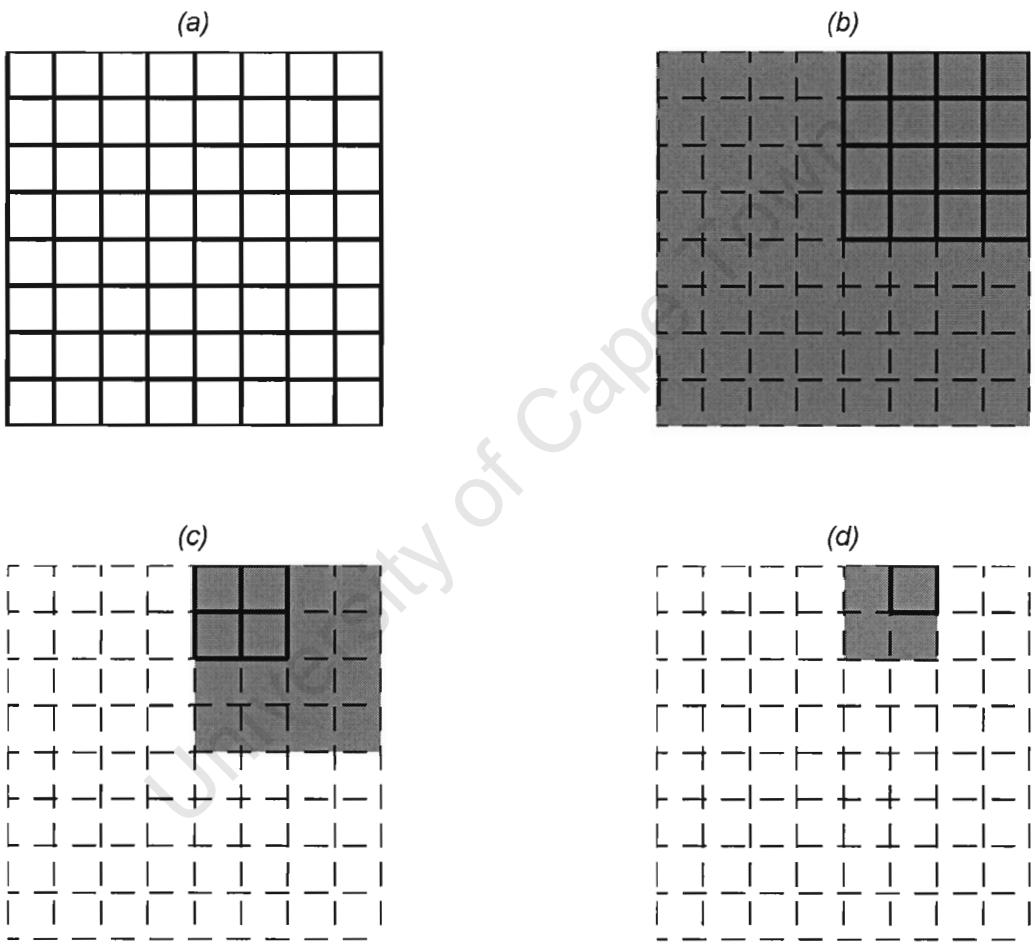


Figure 4.4: Graphically representation of image partitioning
The process is done here for an main image containing $M \times N = 8 \times 8$ pixels, using a factor $F = 3$ to control the image partitioning process thought $S = 4$ stages of processing.

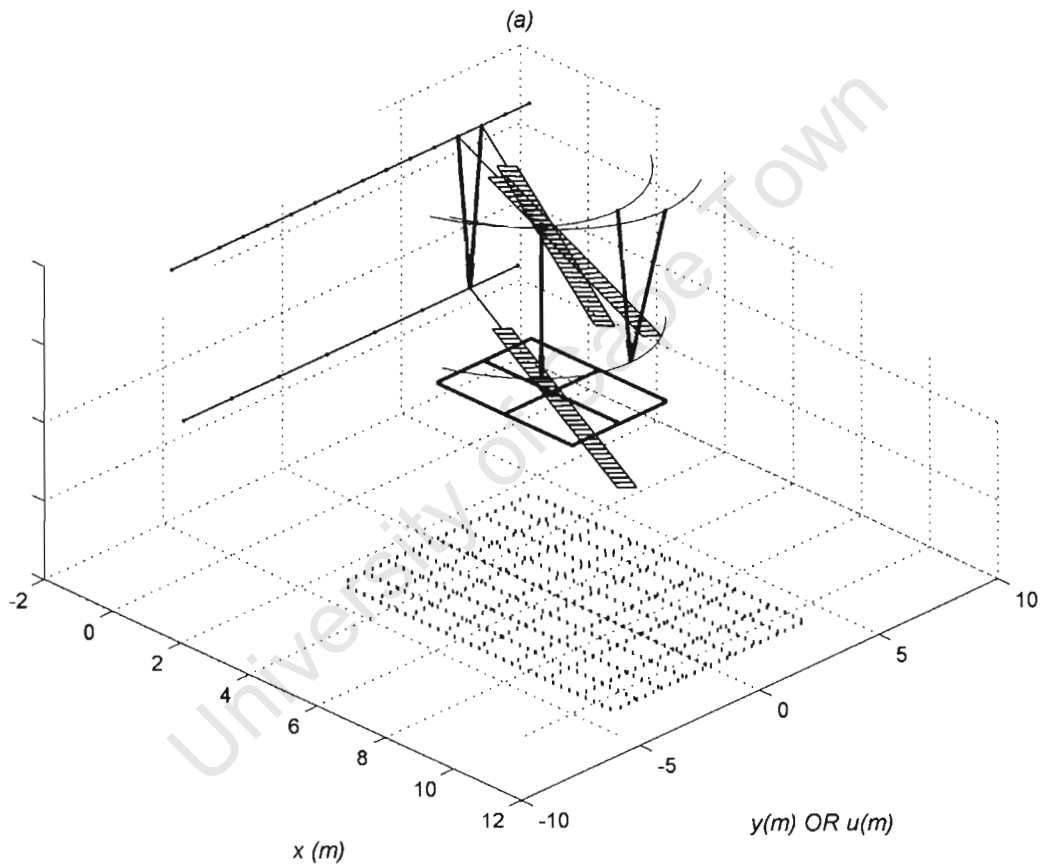


Figure 4.5: New range bin interpolation.
Note that no information is inserted into the sub image grid. The grid is only used for interpolation purposes

new aperture $\overrightarrow{u_s}[s+1]$. Downrange profiles accumulation could be thought of as a $2D$ matched filtering process, which amplifies the pulse echo data for points within the sub-image region in preference to pulse echo data of points outside of the sub-image to produce $K_L[s]$ downrange profiles from the $L_K[s]$ downrange profiles. Thus the downrange profiles accumulation process extracts the pulse echo information relevant to focusing pixels in the sub-image and ignores pulse echo data outside of the sub-image.

- To calculate the range bin values of the new downrange profiles, the downrange profiles are overlaid on the sub-image so that they pass through the sub-image centre (Figure 4.2 (d) (ii) and (iv)).
- To produce this new downrange profile, the interpolation must be made for

$$O_{MN}[s] = M_F[s] \tag{4.19}$$

  points, each spaced dR away from one another.
- These points must be positioned around the sub-image centre, on a diagonal line that lies on the $x-y$ plane, which is a projection of the line that passes through the sub-aperture and sub-image centre.

The line on which all the backprojection points are situated is given by :

$$\langle x_R, y_R\rangle[o_{mn}] = \rho_o[o_{mn}]\langle\cos(\theta_R[k_l]), \sin(\theta_R[k_l])\rangle + \langle u_x, u_y\rangle \tag{4.20}$$

where $\langle u_x, u_y, u_z\rangle = \overrightarrow{u_s}[s+1, k_l]$ is the sub-aperture centre, $\langle x_c, y_c, 0\rangle$ is the sub-image centre. The angle $\theta_R[k_l]$ which the line makes with the x axis is given by:

$$\theta_R[k_l] = \arg\frac{u_y - y_c}{u_x - x_c}. \tag{4.21}$$

The distance $\rho_o[o_{mn}]$ from point $\langle u_x, u_y\rangle$ on the $x-y$ plane is given by :

$$\rho_o[o_{mn}] = \left(o_{mn} - \frac{O_{MN}[s]+1}{2}\right)dR + \rho_c \tag{4.22}$$

where $\rho_c[s] = |\langle x_c, y_c, 0\rangle - \langle u_x, u_y, 0\rangle|$ is the distance of the projected line on the $x-y$ plane from sub-aperture centre to sub-image centre.

The variables $(\rho_o[o_{mn}], \theta_R[k_l])$ are local polar coordinates $[1, 2]$ used to index the proper points on the $x-y$ plane. Thus the intermediate stages in FFBP use local polar coordinates to produce the range loci.

The value of the new range bin is given by :

$$g_{s+1,i_f,j_f}[o_R, k_l] = \left[\sum_{l_k=1}^{F} G(\overrightarrow{r_R}, \overrightarrow{u}[s, l_k])\, g_{s,i_m,j_n}[o_f, l_k]\, R_f e^{j4\pi\kappa_c R_f}\right] e^{j4\pi\kappa_c R_o} \tag{4.23}$$

where

$$\overrightarrow{r_R} = \langle x_R, y_R, H(x_R, y_R)\rangle \tag{4.24}$$

is the target position in space used to calculate the range loci, o_f the indexing variable for the range bin along the target range locus R_f and $[i_m, j_n] = \left[\frac{i_f}{F}, \frac{j_n}{F}\right]_{CEIL}$ is the index of the unfocused SAR data and corresponding sub-image of the previous stage. It can be seen that each new range bin is produced from F range bin values from the previous stage which are situated along the range locus given by R_f. The phase factor $e^{j4\pi\kappa_c R_o}$ is inserted because the new interpolated data must be representative of basebanded unfocused SAR data.

The range from sub-aperture point to target point is given by:

$$R_f = |\overrightarrow{r_R} - \overrightarrow{s_u}[s, k_l, l_k]| \tag{4.25}$$

and the distance between target point and the sub-aperture centre where the new calculated range bin value will be placed is given by :

$$R_o = |\overrightarrow{r_R} - \overrightarrow{u_s}[s+1, k_l]| \tag{4.26}$$

The graphical representation of this process is given in Figure 4.5.

Note The new range bin index in equation 4.23 is labelled as o and not o_{mn}. This is because the uneven ground $H(x_R, y_R)$ does not always allow the new range bin to be evenly distributed. Only when focusing in slant range will $o = o_{mn}$. This is not illustrated in Figure 4.5.

4.2.6 Final Stage Focusing

The final focusing of a sub-image in the final stage to a $F \times F$ image grid is similar to equation 4.4. Because the position of a image pixel in stage $S-1$ is the same as the centre of the 1×1 pixel image in stage S, the point used to calculate the locus of the phase centre is given by:

$$\overrightarrow{r_{ij}}[m_i, n_j] = \langle x_p[m_i], y_p[n_j], H(x_{p,}, y_{p,})\rangle \tag{4.27}$$

A focused pixel of the sub-image in stage S is given by :

$$h_S[i_f, j_f] = \sum_{l_k=1}^{F} G(\overrightarrow{r_{ij}}, \overrightarrow{s_u}[s, k_l, l_k])\, g_{S-1,i_f,j_f}[o_p, l_k]\, R_{nm} e^{j4\pi\kappa_c R_{nm}} \tag{4.28}$$

$$= h_{S-1,i_m,j_n}[, m_i, n_j] \tag{4.29}$$

where

$$R_{nm} = |\overrightarrow{r_{ij}} - \overrightarrow{s_u}[s, k_l, l_k]| \tag{4.30}$$

is the range from sub-aperture points to target position and o_p is the indexing variable for the interpolated range bins.

The images are focused because the process of downrange profile accumulation produced a set of raw SAR data which is specific to focusing the sub-image in question. The interpolated downrange profiles represent an approximation of pulse echo data as if it was captured at the F interpolated

aperture points in stage S, and not L aperture points as in stage $s = 0$. Also, the pulse echo data in these new interpolated downrange profiles seems as if it only came from targets in the sub-image region.

Note Note that all downrange profile have the same number of range bins, O . Thus the index of variables o_R, o_{mn}, o_f and o_p can in general be calculated by:

$$o_{mn} = \left[\frac{R_{mn} - R_I}{dR} + \frac{1}{2}\right] \tag{4.31}$$

$$o_f = \left[\frac{R_f - R_I}{dR} + \frac{1}{2}\right] \tag{4.32}$$

$$o_p = \left[\frac{R_p - R_I}{dR} + \frac{1}{2}\right] \tag{4.33}$$

where R is the range, R_I is the position of the first range bin in the unfocused SAR data and dR is the range bin spacing.

4.2.7 Focusing Restrictions

To ensure the accurate focusing of a pixel, the calculation of the locus which corresponds to the range locus must be within $\frac{\delta R}{2}$ of the real range. [2] gives a equation that estimates the maximum range error for all the approximations made when processing a sub-image, given by:

$$|\Delta R|_{max} \approx \frac{D_L D_N}{4R_{min}} \tag{4.34}$$

where D_L is the sub-aperture length in metres, D_N is the sub-image length on the y axis in metres and the distance to the nearest range for focusing a pixel in the sub-image. Thus if $|\Delta R|_{max} \leq \left|\frac{\delta R}{2}\right|$, then the images can be reconstructed. For FFBP this must be obeyed for all stages of processing to produce a focused pixel.

4.2.8 Instruction Count

From the above analysis equations 4.11, 4.14, 4.15, and 4.19 can be used to derive the instruction count for FFBP where $L = F^S$. The complete instruction count $I_C(L, S)$ for a main aperture of L points and main image of $M \times N$ pixels using S stages of processing for all branches as explained above is be given by :

$$I_C(L,S) = \sum_{s=1}^{S} K_L[s] \times I_F[s] \times J_F[s] \times L_F[s] \times O_{MN}[s] \tag{4.35}$$

$$= \sum_{s=1}^{S} LF^{-s} \times \frac{M}{F^S}F^s \times \frac{N}{F^S}F^s \times F^1 \times F^{S-s} \tag{4.36}$$

$$= \sum_{s=1}^{S} F^{S-s} \times \frac{M}{F^S}F^s \times \frac{N}{F^S}F^s \times F^1 \times F^{S-s} \tag{4.37}$$

$$= MNF\sum_{s=1}^{S} \tag{4.38}$$

$$= MNFS \tag{4.39}$$

$$= MNF \log_F L \tag{4.40}$$

where $F = \sqrt[S]{L}$, since the initial aperture points L are related to the factor F and stages S by equation 4.8. In each stage of processing, each of the $K_L[s]$ sub-apertures will produce a downrange profile for the $I_F[s] \times J_F[s]$ new sub-images by backprojecting $L_F[s]$ sub-aperture points for each of the $O_{MN}[s]$ new range bins. It can be seen that the factorisation constant which is a number of aperture points in each sub-aperture $L_F[s] = F$, is constant throughout the progression of the algorithm.

From equation 4.39 it can be seen that each stage will contain a constant amount of MNF instructions, and equation 4.40 shows that the process is proportional to $\mathcal{O}(n^2 \log n)$. This can also be shown if $L = M = N = F^S$:

$$I_C(L, S) = F \log_F(L) NM \tag{4.41}$$
$$= F \log_F(L) L^2 \tag{4.42}$$

If $S = 2$ then $F = \sqrt{L}$ and the instruction count is:

$$I_c(L, 2) = F \log_F(L) MN \tag{4.43}$$
$$= \sqrt{L} \log_{\sqrt{L}}(L) MN \tag{4.44}$$
$$= \sqrt{L} 2MN \tag{4.45}$$
$$= 2L^{\frac{1}{2}} MN \tag{4.46}$$

This is the instruction count of the LBP algorithm. If $S = 1$, then $F = L = F^1$, and the FFBP algorithm performs one-stage filtered backprojection with instruction count:

$$I_c(L, 1) = F \log_L(L) MN \tag{4.47}$$
$$= L(1) MN \tag{4.48}$$
$$= LMN \tag{4.49}$$

The algorithm's theoretical optimum, which requires the least amount of instructions to reconstruct an image, is reached when $F = e$ [1] and $S = \ln(L)$, is given by :

$$I_c(L, \ln L) = F \log_L(L) MN \tag{4.50}$$
$$= e \ln(L) MN \tag{4.51}$$

4.2.9 FFBP Computational Saving

The theoretical saving of the FFBP algorithm can be seen when compared to the normal one stage filtered backprojection. The instruction count for one stage backprojection is LMN. The

instruction count for FFBP is $I_C(L,S) = MNF\log_F L$. The saving and ratio of time spent on inversion of FFBP in comparison to FBP is given by:

$$\begin{aligned} I_S(L,S) &= \frac{LMN}{MNF\log_F L} && (4.52)\\ &= \frac{L}{F\log_F L} && (4.53)\\ &= \frac{L}{FS} && (4.54)\\ &= \frac{F^S}{FS} && (4.55)\\ &= SF^{S-1} && (4.56) \end{aligned}$$

Equations 4.52 show that the saving is independent of the $M \times N$ pixel size of the image. If $S = 2$ then $F = \sqrt{L}$ and the saving is given by:

$$\begin{aligned} I_s(L,2) &= \frac{F\log_F L}{L} && (4.57)\\ &= \frac{\sqrt{L}2}{L} && (4.58)\\ &= 2\sqrt{L} && (4.59) \end{aligned}$$

Even using two stages of processing gives a huge saving in comparison to the one stage FBP. To show that using one stage of processing does not have any computational benefit in terms of saving, where stages $S = 1$ and $L = F^S = F^1 = F$, is given by:

$$\begin{aligned} I_S(L,1) &= \frac{F\log_F L}{L} && (4.60)\\ &= \frac{L(1)}{L} && (4.61)\\ &= 1 && (4.62) \end{aligned}$$

For $L = F^S = e^{\ln L}$ and $S = \ln L$ the optimal theoretical saving due to the reduction of instructions is :

$$\begin{aligned} I_S(L,\ln L) &= \frac{F\log_F L}{L} && (4.63)\\ &= \frac{e\ln L}{L} && (4.64)\\ &= \frac{S}{e^{S-1}} && (4.65) \end{aligned}$$

4.3 Local Backprojection (LBP)

LBP is explained in [3] and uses two stages of processing to focus SAR images. It has an operational count proportional to an $\mathcal{O}(2n^{2.5})$ process, which is significantly less than the FBP which has an operational count proportional to an $\mathcal{O}(n^3)$ process. For focusing large apertures of say

4096 downrange profiles to an image size of 2048×2048 pixels a theoretical saving of 32 times can be achieved. The LBP can be interpreted as a FFBP algorithm using two stages of processing that process data in a specific way.

Because LBP only uses two stages of processing, however there are a few practical advantages which can be achieved. The second stage of LBP SAR inversion is implemented in a different way in [3] than in [2].

1. Stage 1 and stage 2 are interleaved instead of processing stage 2 after the completion of stage 1.
2. In the second stage of processing, backprojection is not done by weighted coherent summing but by adding the range bin value to the image by spreading the information over the circular locus with radius corresponding to the range bin range.
3. By doing this the value of the downrange profile accumulated can be used immediately without the necessity of storage. Thus the range bins which 2 above is referring to are virtual.

Item 3 above has a very useful trade off for saving storage resource. This will be discussed in Chapter 5. Other than this, there are two extra approximations which do not reduce the instruction count but simplify the complexity of reconstructing the SAR image. The drawback of the algorithm is that it cannot focus accurately to ground range, due to the way stage two is processed. If Item 2 above is processed in the normal way by weighted coherent summing, then the algorithm can focus to ground range as explained in Section 4.2.6. However, the downrange profile accumulated values will have to be stored.

4.3.1 LBP Approximations

To simplify the geometrical complexity of SAR processing, two approximations can be drawn from filtered backprojection for stage one and stage two respectively [3].

4.3.1.1 Approximation No 1

In one-stage backprojection (FBP) a pixel at image co-ordinate $[m, n]$ must first be mapped to the world co-ordinate system $[m, n] \Rightarrow \langle x_p, y_p \rangle$. The range locus, along which backprojection will take place, can be calculated from the co-ordinate $\langle x_p, y_p \rangle$ in order to focus the pixel at image co-ordinate $[m, n]$. On close inspection it is observed that the range locus passing over the sub-apertures can be approximated by straight lines instead of a curve. Thus, there will be a straight line in each sub-aperture which will approximate a part of the range locus. This can be seen when placing the range locus on the multi-aperture grid as in Figure 4.6 (a), (b) and (c). This results in a less complex representation of the range loci. The curve will more accurately represent a straight line the further a target is positioned from the radar. This approximation pays off very well if most of the targets are situated very far from the radar. Phase history signatures far from the radar usually produce loci which are almost straight. This favours the fast algorithm making it more accurate for targets in the far field. Instead of now integrating along the range locus, backprojection of one point-target is performed as a collection of integrals along straight lines.[3]

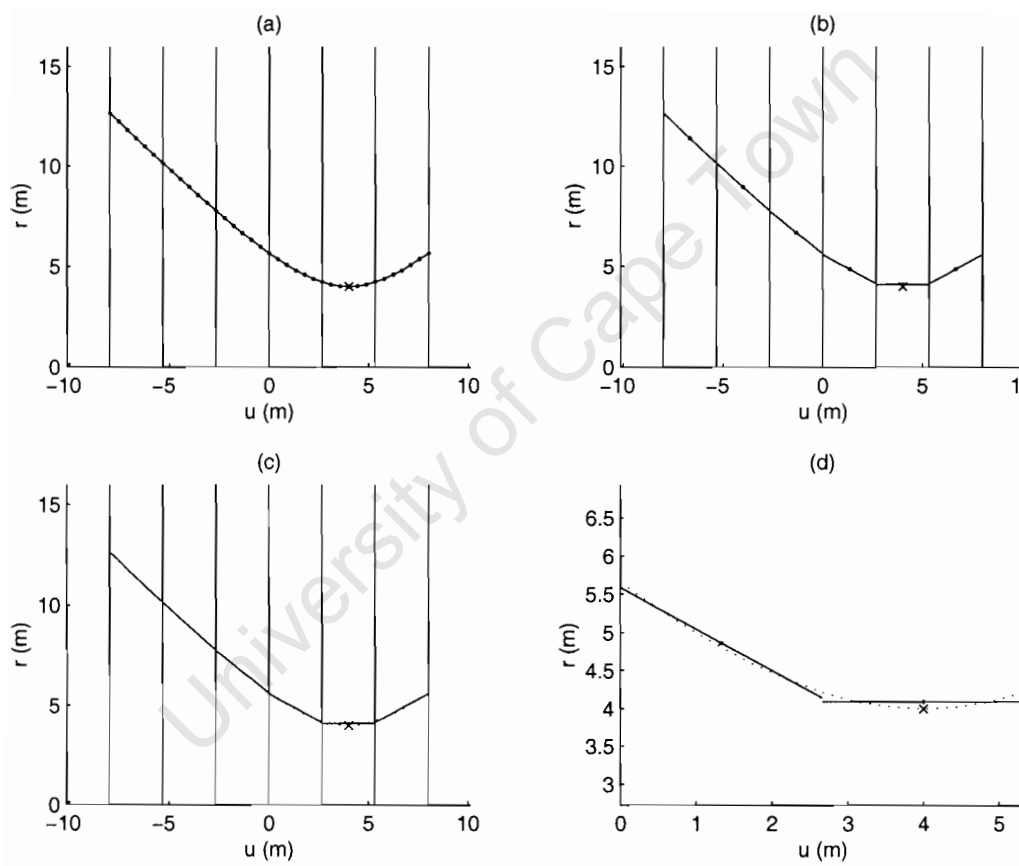


Figure 4.6: Approximation of hyperbolic loci by lines

4.3.1.2 Approximation No 2

In stage two, the value of a point at $\langle r, u \rangle$ in the $r-u$ plane will represent a circle in the $x-y$ plane with centre at $\langle 0, u \rangle$ and having a radius r. Again, by placing the circle on a multi-image grid as in Figure 4.7, it can be seen once more that the curve through a sub-image can be approximated by a straight line. Thus, the further the target is from the radar the greater the radius of the circle and therefore the more accurately a straight line can represent the locus of the circle on the sub-image [3].

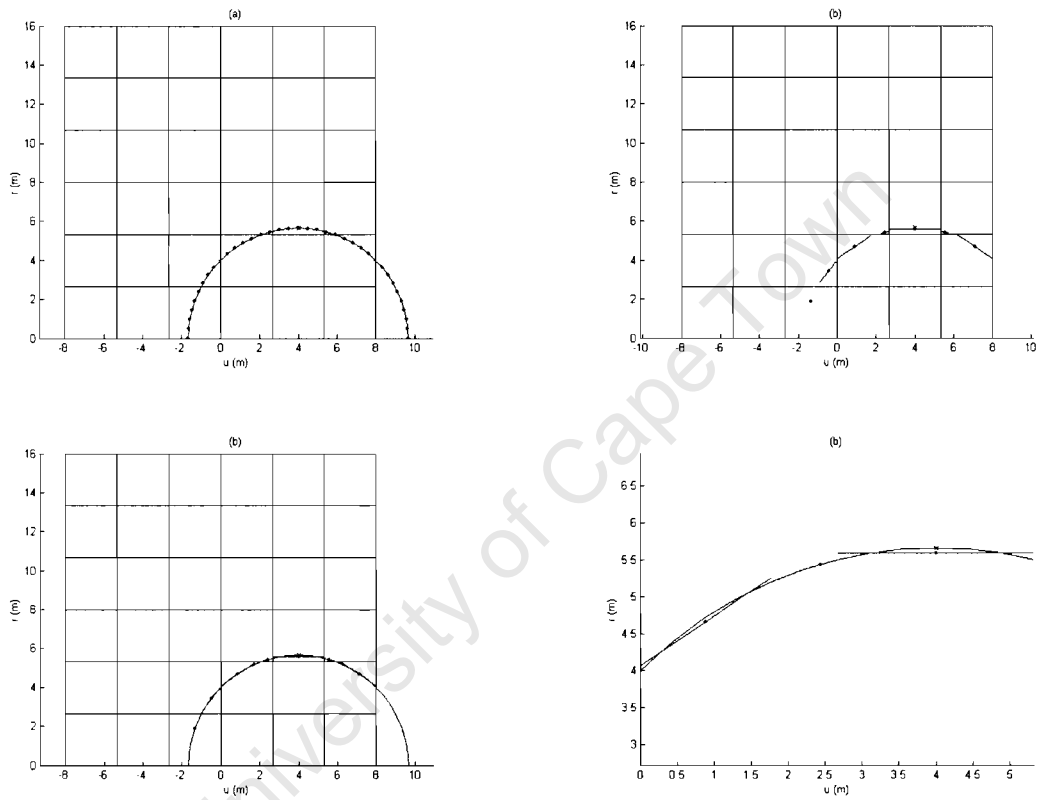


Figure 4.7: Approximation of circular loci by lines

Although the above two approximations describe a simplification of the processing of SAR focusing, it does not reduce the instruction count of the algorithm.

Chapter 5

Implementation of SAR Algorithms

In order to compare the quality of images produced by FFBP to FBP, the thesis project required the implementation of a one-stage filtered backprojection and a multi-stage fast factorised filtered backprojection SAR image focusing processor. In addition, the unfocused SAR data were produced by a simulator developed for the purpose of the thesis. The simulator produces unfocused SAR data which is a discrete version of the radar echo model of Chapter 2. This provides the unfocused SAR data which would otherwise have been digitally acquired by a wide beam, wide bandwidth radar. This chapter deals with the practical details of implementing these three programs.

Issues like data storage and data representation are described. In FFBP the choice of loop sequence and use of resources for storing data used in processing intermediate stages, are interlinked. The choice and trade offs of the manner in which the FFBP algorithm implements its loop sequence are discussed. The practical issues of data movement from disk to memory which slow the FFBP algorithm down from its theoretical optimum are also addressed by the implementation of buffering to increase the data throughput.

The simulator was implemented in Matlab. Both image focusing processors which digitally reconstruct SAR images were implemented in both Matlab and C++.

5.1 Implementation Objectives

The two SAR inversion algorithms used to focus SAR data are part of a class of time domain backprojection, ultra wide band focusing algorithms [1, 2] which are capable of inversion of unfocused SAR data which are acquired from a nonlinear flight path using a radar with an antenna which has a wide beamwidth [1, 2]. The implementation of the simulator and SAR digital image reconstruction algorithms must produce and process data which are representative of the above.

Likewise the theory in Chapter 2 uses the assumptions that the unfocused SAR data are acquired by a monostatic radar [1]; the signal travels through a linear propagation medium [1], that the "start stop approximation" applies [14] and the scene consists of a collection of single-scattering point targets (Born approximation) [1, 2]. The SAR Simulator was built to take into account the above assumptions when processing the scene to produce data that would be digitally representative of the radar echo model in Chapter 2.

Some physical factors which were not taken into account when designing the simulator and focusing processors are:

1. The effects of shadowing and layover
2. Real SAR sensing is done with an air borne vehicle which can roll, yaw and pitch role causing unforeseen changes in phase that must be taken into account for SAR simulation.
3. The information about the transmitted pulse; such as pulse length or spectral components.
4. The incidence angle of the pulse to the ground is not taken into account
5. The effect of polarisation due to the antenna orientation
6. The targets may also be moving.
7. The RCS of each point on the ground has a frequency spectrum which will not attenuate reflect signals of different frequencies in the same way.

These factors were neglected because the thesis project concentrates on focusing whereas these issues are more important to the accurate simulation of SAR data. Some assumptions made and reasons for neglecting these designed considerations are given below.

1. Although the effect of shadowing and layover can be simulated, it was not part of the objectives to study the effect of shadowing and layover when focusing. Thus it was left out of the design of the simulator.
2. The effect of roll, yaw and pitch of the airborne vehicle is not taken into account for SAR image focusing as it is to complex for our simple investigation. Curved trajectories were however modelled.
3. The simulator produces a basebanded range compressed output and the focusing algorithms use this as input. Thus a complex frequency spectrum is not needed to simulate or focus the SAR data. Only the bandwidth and centre frequency, wave speed and signal amplitude at transmission are needed.
4. It is assumed by the radar echo model that the tilt angle of the radar is at zero throughout the simulation and focusing.
5. It is assumed by the radar echo model that the ground targets are always stationary.
6. It is assumed by the radar echo model that the signal response of a point target is uniform for all frequency.

5.2 Simulator and Focusing Processor Design

Sections 5.2, 5.2 and 5.4 contains information relevant to implementation of the simulator and digital image reconstruction programs. These sections are intended for programmers wishing to implemented the programs mentioned in this thesis. Detail descriptions for indexing information in 2D array data sets are also given.

Both the simulator and image focusing processors requires as input, a single setup file and a number of two dimensional(2D) data sets. Both the simulator and image focusing processors

will produce a single 2D data set as output data. In the case of the simulator this set of data would represent the digitally acquired unfocused range compressed SAR data sampled down to baseband. In the case of the SAR inversion algorithms the output will be representative of a ground focused image on the $x - y$ plane.

The setup file contains all information to retrieve input files as well as the output paths to store data, to either simulate or focus data depending on the program used. The setup file contains information regarding:

1. The transmitted signal waveform (TSW) e.g. bandwidth and centre frequency
2. The radar flight path (RFP) information
3. The ground reflectivity map (GRM) of the ground surface
4. A digital elevation model (DEM) of the ground and
5. Antenna beam pattern (ABP) of the radar

Figure 5.1 shows the data flow of information from input files to focused image.

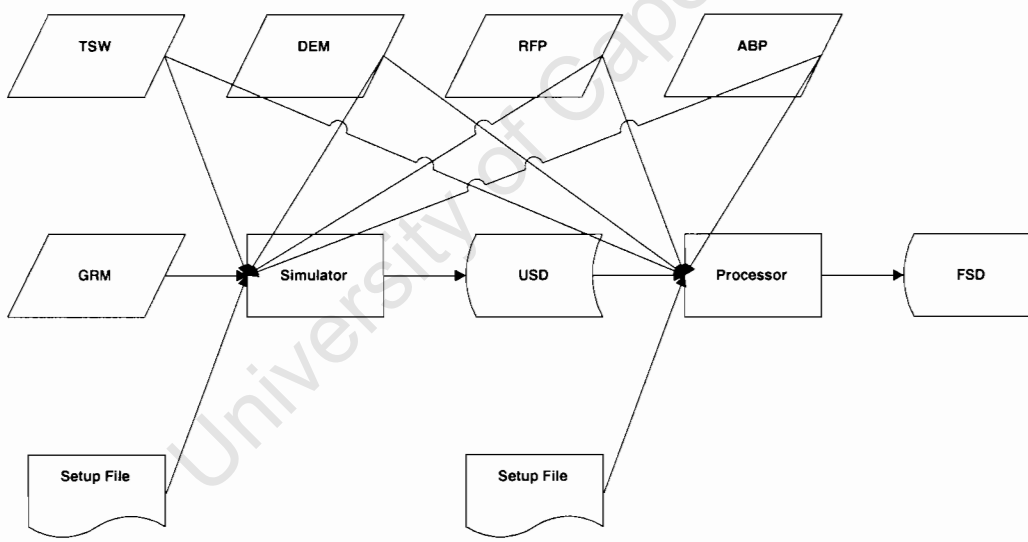


Figure 5.1: Information flow of SAR simulator & image focusing processor

The diagram shows how input information contributes to both simulator and image focusing processor. The top row shows common input information which aid in both simulation and focusing. The simulator takes in a ground reflection map (GRM) and the image focusing processor takes in unfocused SAR data (USD). Both programs also require independent setup files.

5.2.1 Data Storage Representation

All input and output data sets represent 2D data similar to matrices. The data sets are divided into rows and columns. Information is stored so that all variables (types) within a column are stored adjacent to one another and all rows are the stored adjacent to one another. In the case of complex data the real part is stored before the imaginary part. The information storage uses two main data types, 32 bit precession floating point variables (float) and 64 bit double precision floating point

variables (double). The Matlab versions can only process doubles whereas the C++ version can be compiled to process floats or doubles.

All 2D data sets are binary. Both the simulator and focusing processors only read in as input and produce as output binary data as specified above. If the input data is not in the above specified format, a preprocessor program is required to transform the data. Likewise a post processor is required to transform any output data into any other format. e.g. pgm, ps or jpeg image files.

5.2.2 Input Data Formats

For both simulator and processor there are three input data sets used to reference information. They are :

1. Radar flight path (RFP)
2. Digital elevation model (DEM)
3. Antenna beam pattern (ABP)

The ground reflectivity map(GRM) is only used by the simulator as input to produce the unfocused SAR data.

5.2.2.1 Radar Flight Path(RFP)

The radar flight path contains the information described by equation 4.1. The radar flight path consist of a $3 \times L$ long array of either double of float type variables. The information is stored so that there are L rows containing 3 columns. Each of the rows represent an aperture point in space. Each point of the aperture position vector, $\langle x, y, z \rangle$, is represented by a separate column. Thus the flight path takes the form of equation 4.1 which has a finite length consisting of L vector points.

5.2.2.2 Digital Elevation Model (DEM)

The DEM is a digital representation of the height map $H(x, y)$ described by :

$$H[i_x, j_y] = H(x[i_x], y[j_y]) \tag{5.1}$$

where $[i_x, j_y]$ are the indexing variables with cell spacing (dx, dy). The DEM consists of an $I_X \times J_Y$ long array of either double or float type variables. The information is stored so that there are J_Y rows containing I_X columns. The rows represent the $\hat{\mathbf{y}}$ direction and the columns represent the $\hat{\mathbf{x}}$ direction. The orientation of each cell centre in the $x - y$ plane is given by :

$$x[i_x] = \left(i_x - \frac{I_X + 1}{2}\right) dx + X_I \tag{5.2}$$

and

$$y[j_y] = \left(j_y - \frac{J_Y + 1}{2}\right) dy + Y_J \tag{5.3}$$

where $\langle X_I, Y_J \rangle$ is the centre of the DEM on the $x - y$ plane. The variables $[I_X, J_Y]$, (dx, dy) and $\langle X_I, Y_J \rangle$ are found in the setup file. The reference height of an $\langle x, y \rangle$ co-ordinates are found

by inverting equations 5.2 and 5.3. This gives the cell index used to reference the height from the ground co-ordinates $\langle x, y \rangle$ on the $x - y$ plane:

$$i_x(x) = \left[\frac{x - X_I}{dx} + \frac{I_X + 1}{2} \right] \tag{5.4}$$

and

$$j_y(y) = \left[\frac{y - Y_J}{dy} + \frac{J_Y + 1}{2} \right] \tag{5.5}$$

5.2.2.3 Antenna Beam Pattern (ABP)

The antenna beam pattern gives a discrete representation of the distribution $G(\theta_A, \phi_E)$ described by:

$$G[i_\theta, j_\phi] = G(i_\theta d\theta_A, j_\phi d\phi_E) \tag{5.6}$$

where $[i_\theta, j_\phi]$ are the indexing variables with cell spacing $(d\theta_A, d\phi_E)$. The antenna beam pattern consist of an $I_\Theta \times J_\Phi$ long array of either double or float type variables. The information is stored so that there are J_ϕ rows containing I_Θ columns. The rows represent the θ_A axis and the columns represent the ϕ_E axis in spherical co-ordinates. The position of the angles (θ_A, ϕ_E) in relation to the indexing variables $[i_\theta, j_\phi]$ on the $\theta_A - \phi_E$ spherical surface is given by :

$$\theta_A[i_\theta] = \left(i_\theta - \frac{I_\Theta + 1}{2} \right) d\theta_A + \Theta_A \tag{5.7}$$

and

$$\phi_E[j_\phi] = \left(j_\phi - \frac{J_\Phi + 1}{2} \right) d\phi_E + \Phi_E \tag{5.8}$$

where $\langle \Theta_A, \Phi_E \rangle$ is the centre of the antenna beam pattern on the $\theta_A - \phi_E$ plane in spherical co-ordinates. The variables $[I_\Theta, J_\Phi]$, $(d\theta_A, d\phi_E)$ and $\langle \Theta_A, \Phi_E \rangle$ are found in the setup file. The cell index used to reference the antenna gain, given the angle$\langle \theta_A, \phi_E \rangle$ a target makes with the radar boresight is given by:

$$i_\theta(\theta_A) = \left[\frac{\theta_A - \Theta_A}{d\theta_A} + \frac{I_\Theta + 1}{2} \right] \tag{5.9}$$

and

$$j_\phi(\phi_E) = \left[\frac{\phi_E - \Phi_E}{d\phi_E} + \frac{J_\Phi + 1}{2} \right] \tag{5.10}$$

NB: The antenna beam pattern distribution $G[i_\theta, j_\phi]$ must take into account the cosine scaling factor of $\cos(\pi\phi_E)$ in the ϕ_E direction.

5.2.2.4 Ground Reflectivity Map (GRM)

The GRM $\zeta(x, y)$ is described by:

$$\zeta[i, j] = \zeta(i dx_i, j dy_j) \tag{5.11}$$

The GRM consist of an $I \times J$ long array of either double or float type variables. The information is stored so that there are J rows containing I columns. The rows represent the $\hat{\mathbf{x}}$ direction and the columns represent the $\hat{\mathbf{y}}$ direction. The orientation of each cells centre on the $x-y$ plane in both $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ direction is given by:

$$x_i[i] = \left(i - \frac{I+1}{2}\right) dx_i + X_\sigma \tag{5.12}$$

and

$$y_j[j] = \left(j - \frac{J+1}{2}\right) dy_j + Y_\sigma \tag{5.13}$$

where $[i,j]$ are the indexing variables with pixel spacing (dx_i, dy_j) of the $[I,J]$ points in the respective $\langle \hat{\mathbf{x}}, \hat{\mathbf{y}} \rangle$ directions and $\langle X_\sigma, Y_\sigma \rangle$ is the image grid centre. The variables (dx_i, dy_j), $[I,J]$ and $\langle X_\sigma, Y_\sigma \rangle$ are found in the setup file. Using equations 5.4, 5.5, 5.12 and 5.13 the position of a point $\overrightarrow{r}$ with attenuation coefficient $\zeta[i,j]$ is found by:

$$\overrightarrow{r}[i,j] = \langle x_i[i], y_j[j], H[i_x(x_i[i]), j_y(y_j[j])] \rangle \tag{5.14}$$

Thus the output of the SAR simulator will only be representative of all GRM data which is found on the DEM.

The GRM map can be used by the radar echo model of equation 2.74 by setting.

$$k = Ji + j \tag{5.15}$$

Then $\overrightarrow{r}[k] = \langle x_i[i], y_j[j], H[i_x(x_i[i]), j_y(y_j[j])] \rangle$ and the GRM can be described as:

$$\zeta[k] \quad = \quad \zeta(\overrightarrow{r}[k]) \tag{5.16}$$

Although the GRM gives attenuation coefficients of the centre point in a cell, it is assumed in this implementation that points close to one another all have approximately the same attenuation coefficients. Thus a square patch of ground (dx_i, dy_j) around $\overrightarrow{r}[i,j]$ is assumed to be approximately equal to $\zeta[i,j]$.

5.2.2.5 Transmitting Signal Waveform (TSW)

Only unfocused baseband range compressed SAR data is produced by simulation or needed for processing. Therefore a complete frequency spectrum over the transmitted pulse's bandwidth is not specified. Only the bandwidth and carrier frequency are needed for simulation and focusing. The speed (c), bandwidth (Δf), carrier frequency (f_c) and transmission amplitude (V_t) are specified in the setup file. Information such as PRI, pulse staring time and pulse duration were not explicitly modelled.

5.3 Unfocused SAR Data Simulator (USD)

The unfocused SAR data is a digital representation of the unfocused range compressed SAR data $g(t_A, R)$ sampled down to base band, which is used as input data for the SAR inversion algo-

rithms. The unfocused SAR data consist of a $L \times O$ long array of either complex doubles or complex float type variables. The information is stored so that there are L rows containing O columns. The rows represent the azimuth direction and the columns represent the range direction. The output $g[l,o] \approx g(t_A[l], R[o])$ is only an approximation of a real scene. The orientation of each downrange profile and range bin in the projected plan is given by:

$$t_A[l] = \left(l - \frac{1}{2}\right) dt_A + T_I \tag{5.17}$$

and

$$R[o] = \left(o - \frac{1}{2}\right) dR + R_I \tag{5.18}$$

where $[l,o]$ are the indexing variables with pixel spacing (dt_A, dR) of the $[L,O]$ downrange profiles & range bins, and $\langle T_I, R_I \rangle$ is the start of the unfocused SAR data in azimuth time and range. Again the variables (dt_A, dR), $[L,O]$ and $\langle T_I, R_I \rangle$ can be found in the setup file. Each of the range bins in a downrange profile contains information of each of the $I \times J$ points of the $\zeta[i,j]$ GRM map. Thus each range bin is a sum of $I \times J$ values given by :

$$g[l,o] = \sum_{j=1}^{J} \sum_{i=1}^{I} \zeta[i,j]\, q_{mf}[l,o,\overrightarrow{r}[k]] \tag{5.19}$$

where

$$q_{mf}[l,o,\overrightarrow{r}[k]] \quad = \quad q_{mf}(t_A[l], R[o], \overrightarrow{r}[k]) \tag{5.20}$$

is a discrete approximation of $q_{mf}(t_A, R, \overrightarrow{r}[k])$, the phase history of a ideal point target which reflects all received signal and

$$\zeta(\overrightarrow{r}[k]) \quad = \quad \zeta[i,j] \tag{5.21}$$

is the ground reflectivity map at point $\overrightarrow{r}[k]$ in the scene.

On completion noise can be added. The discrete approximation of the range loci for simulation of a point target is given by:

$$R[k,l] = |\overrightarrow{r}[k] - \overrightarrow{u}[l]| \tag{5.22}$$

and the range bin corresponding to that range loci centre is given by:

$$o \quad = \quad \left[\frac{R[o] - R_I}{dR} + \frac{1}{2}\right] \tag{5.23}$$

5.3.1 Range Bin Population Sequence

The sequence in which range bins are populated does not in any way impact on changing the amount of memory or storage resources being used. The simulation of the unfocused range compressed SAR data is a straight forward one stage algorithm consisting of $I \times J \times L \times O$ loops

which is comparable to an $\mathbf{O}\left(n^4\right)$algorithm. The sequence of loop processing is shown in Figure 5.2.

5.4 Filtered Backprojection Processor

The filtered backprojection processor produces the digitally reconstructed image. The complete description of the digital image reconstruction was given in Section 4.1. The data storage format and loop sequence is discussed here.

The focused SAR data(FSD) is a digital representation the the ground range image $f\left(x,y\right)$ given by:

$$f\left[m,n\right] \quad = \quad f\left(x_m\left[m\right],y_n\left[n\right]\right) \tag{5.24}$$

The focused SAR data consist of a $M \times N$ long array of either complex double or complex float type variables. The information is stored so that there are N rows containing M columns. The rows represent the $\hat{\mathbf{x}}$ direction and the columns represent the $\hat{\mathbf{y}}$ direction.

5.4.1 Focusing Sequence of Pixels

As in Section 5.3.1 the sequence in which image pixels are populated does not in any way impact on changing the amount of memory or storage resources which are used. The focusing of a SAR image by one stage filtered backprojection is a straight forward one stage algorithm consisting of $L \times M \times N$ loops which is comparable to an $\mathbf{O}\left(n^3\right)$algorithm as explained in Section 4.1 . The sequence of loop processing is shown in Figure 5.3.

5.5 Fast Factorised Backprojection Processor

Like Section 5.4 the fast factorised filtered backprojection processor is completely described in Section 4.2. The FFBP also produces a focused SAR data output. Thus the focused SAR data data storage format is the same as given in Section 5.4.

However there are a few practical issues which have to be addressed when implementing the FFBP algorithm. Although Section 4.2.3 describes the implementation of FFBP for discrete variables there are a few restrictions brought on by aperture factorisation and image splitting.

Equation 4.8 gives the relation that $L = F^S$. If both $F \geq 2$ and $S \geq 2$ are integers it means that L does not cover the complete set of natural numbers $\aleph$. Furthermore numbers like $L = 216$ can be factorised by $S = 3$ stages but not $S = 2$ stages. It might be necessary to use fewer number of stages in order to ensure a focused image.

This section describes a relaxation of the constraints placed upon the algorithm. By doing this the process of digital image reconstruction using FFBP becomes more general to focusing an aperture of almost any length. All modification made below must obey the demands of the algorithm which state that:

1. The minimum range tolerance for focusing must be maintained through all stages [1, 2]. i.e. to obtain a focused pixel this implementation requires $\left|\Delta R\right|_{max} \leq \left|\frac{\delta R}{2}\right|$ for all stages of focusing.

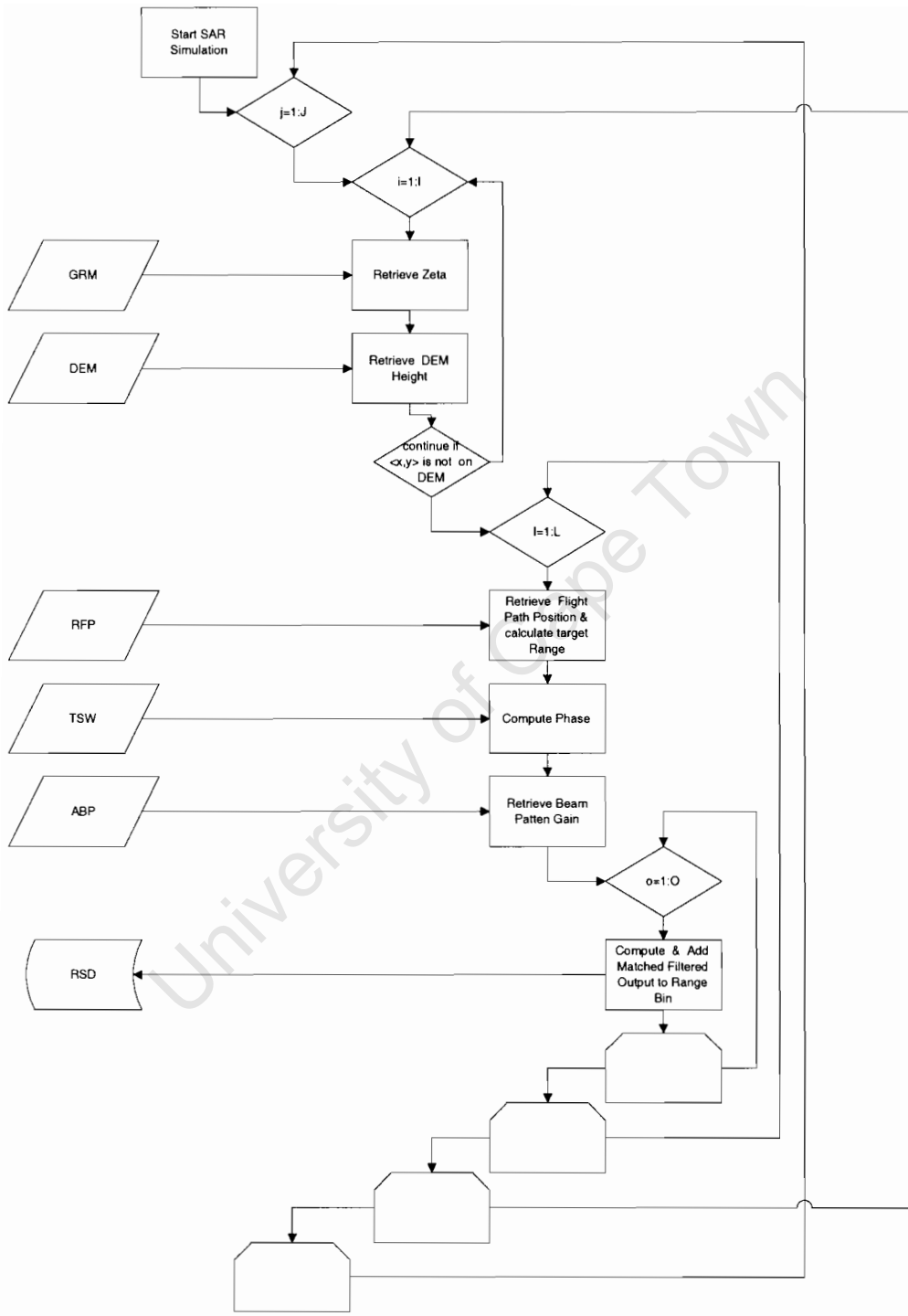


Figure 5.2: Flow diagram of the simulation of unfocused basebanded range compressed SAR data. The diagram is intended to show the four loops needed to produce the simulated data. The sequence of events simulating a range bin briefly shown. The ground reflectivity map, DEM, radar flight path, Transmitted signal information and antenna beam pattern all aid in producing the bin. Output is stored in the unfocused SAR data array.

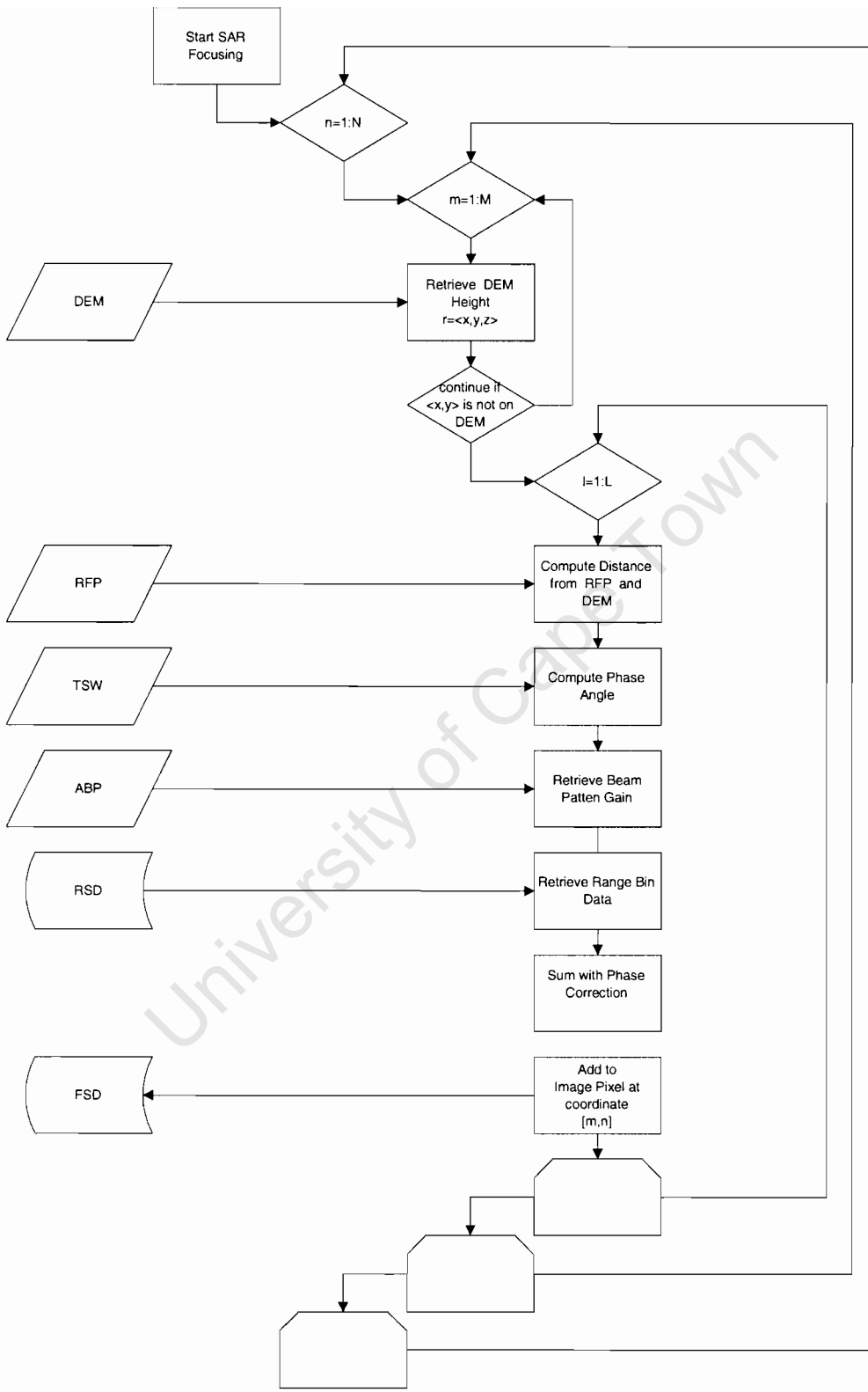


Figure 5.3: Flow diagram of one stage filtered backprojection .

Instead of ground reflectivity map (GRM) the unfocused SAR data (USD) is the main input in the flow diagram. The diagram is intended to show the three loops needed to produce the focused data. The sequence of events focusing a pixel is briefly shown. The unfocused SAR data, DEM, radar flight path, transmitted signal information and antenna beam pattern all aid in producing the pixel. Output is stored in the focused SAR data array.

2. The final stage ends with each of the images being 1×1 pixels in dimension. [1, 2].

The modifications to the FFBP algorithm which will be described are only valid if the above are obeyed.[1]

5.5.1 Factorisation Problem

As described in [1] a general description of the number of apertures points can be given by:

$$L = f_1 \times f_2 ... \times f_{S-1} \times f_S$$

In a discrete representation where each of the f_s variables are positive integers, the maximum number of stages S_L will be equal the total number of prime factors of L. This will allow all positive non primes to be focused with at least two stages. However this does not allow the data to be focused by any number of stages $S < S_L$. It is also much harder to ensure that $|\Delta R|_{max} \leq \left|\frac{\delta R}{2}\right|$ for factors f_s which are not approximately constant through all stages of processing.

After studying the focusing example in [2] a least restrictive scenario is to allow $f_s = F$ for stages $(1 \leq s \leq S-1)$ to be part of the $\Re^+$.The factorisation of an aperture can then be given by :

$$L = \left(\prod_{s=1}^{S-1} F\right) \times f_S \tag{5.25}$$

$$= \left(\prod_{s=1}^{S-1} F\right) \times f_l F \tag{5.26}$$

where f_S in the final stage S is given by:

$$f_S = \frac{L}{F^{S-1}} \tag{5.27}$$

$$= \frac{L}{F^S} F \tag{5.28}$$

and

$$f_l = \frac{L}{F^S} \tag{5.29}$$

$$= LF^{-S} \tag{5.30}$$

Thus if L and S are integers then the factorisation process can have any number of stages as long as:

1. $(2 \leq F \leq L)$ for stages $(1 \leq s \leq S-1)$. Since F cannot be negative or greater than L.
2. $S < S_L$ where

$$S_L = [\log_2 L]_{FLOOR} \tag{5.31}$$

since a digital reconstruction of a final image can only be focused if there there are two or more downrange profiles.

[1] Failure to obey 1 above may not allow the image to focus . The 2nd consideration must be obeyed as it will make practical programing impossible as it is impossible to index fractional parts of a pixel.

3. $f_S \geq F$ in stage $s = S$.

Although the factors F may not be an integer, the size of individual apertures points L_F can be found by dividing each of the indices of each of the aperture points by F and then rounding them up to the next integer. This would produce an array that designates an aperture point to a specific sub-aperture given by:

$$\begin{aligned} K_F \quad [l_k] = & \left[\frac{l_k}{F}\right]_{CEIL} && (5.32)\\ = & \left[\frac{\{1, 2, \ldots, L_K[s] - 1, L_K[s]\}}{F}\right]_{CEIL} && (5.33)\\ = & \{\{1, \ldots, 1\}, \ldots, \{K_L[s], \ldots, K_L[s]\}\} && (5.34)\\ = & \; k_l \end{aligned}$$

where k_l designates the sub-aperture to which an aperture point belongs to. The number of points in these apertures can now be found by:

$$L_F[s, k_l] = \sum_{1}^{L_K[s]} \frac{K_F[l_k]}{k_l} \qquad (5.35)$$
$$(K_F[l_k] == k_l)$$

where $L_F[k_l] \approx F$ if $F \neq \sqrt[S]{L}$ and $L_F[k_l] = F$ if $F = \sqrt[S]{L}$ for stages $(1 \leq s \leq S - 1)$.

5.5.2 Image Splitting Problem

In Section 4.2.4 the main $M \times N$ pixel image in stage $s = 1$ is split into $\frac{M}{F^S}F \times \frac{N}{F^S}F$ images of $F^{S-1} \times F^{S-1}$ pixels each. This method of image splitting ensures that in the final sub-image, $S = s$, the focused information is inserted into $M \times N$ images 1×1 pixel in size. It does not take into account the size of the dimensions of the main $M \times N$ pixel image in relation to the L aperture points of the main aperture.

In the case where $L = F^S$ and where the main image contains pixels $M \leq F^{S-2}$ or $N \leq F^{S-2}$ in size, the first stage $(s = 1)$ of processing would require the main image to be split into $I_M[1] = \frac{M}{F^S}F \geq \frac{1}{F}$ along the x axis or $J_N[1] = \frac{N}{F^S}F \geq \frac{1}{F}$ along the y axis respectively. This situation is impractical as it is impossible to split images into fractional amounts. Therefore the main image size as used in case of Section 4.2.4 the image splitting process governed by equation 4.15 can only be implemented for a main image of where $M \geq F^{S-1}$ and $N \geq F^{S-1}$ pixels. This is so that it can be split by integer values of $I_M[1] \geq 1$ and $J_N[1] \geq 1$. In general no image may be split into fractional parts less that one irrespective of the relationship of F, L and S.

To prevent this, the number of stages must be limited so that the main image in the first stage can be split up into any integer value $I_M[1] \geq 1$ and $J_N[1] \geq 1$ with $M[1] \times N_F[1] \approx F^{S-1} \times F^{S-1}$ pixels in each sub-image. Thus the number of stage for a factor F which is limit by image splitting is given by:

$$S_{MN} \leq [MIN(\log_F M, \log_F N)]_{Floor} \tag{5.36}$$

where

$$M \geq 2F \tag{5.37}$$

$$N \geq 2F \tag{5.38}$$

to ensure that $(2 \leq S \leq S_{MN})$.

Also the above modification of aperture factorisation brings its own problems. Images can only be split by positive integer values. The fractional part of F must be discarded by rounding down. Rounding up may result in stage $S-1$ having images which are less than 1×1 pixel in size. The process of discarded the fractional part and rounding down may however work against the focusing constraint where $|\Delta R|_{max} \leq \left|\frac{\delta R}{2}\right|$ but there is nothing that can be done.

Thus the factorisation of an aperture also affects the splitting of an image and the maximum number of stages the algorithm can process. To ensure that all images in all stages are not split by fractional numbers, images must be split by numbers only containing integer values greater than one with no fractional parts.

To ensure that all images contain an integer number of pixels, image splitting uses a process which is similar to equation 5.32 to designate image pixels to sub-images. Also the value of $O_{MN}[s]$ must be equal to $M_F[s]$ to ensure that no focusing information is lost in all stages .

5.5.3 Maximum Number of Processing Stages

Looking at the constraints of equation 5.31 and 5.36 the maximum number of stages that can be processed by this method of implementation is:

$$S_F \leq [MIN(S_L, S_{MN})]_{Floor} \tag{5.39}$$

where $(2 \leq F \leq L)$.

5.5.4 General Instruction Count

For the purpose of deriving a general instruction count where the variables F, L, M, N, and S are not related and where the number of stage $S \leq S_F$, the following two variables are introduced.

$$f_m = \frac{M}{F^S} \tag{5.40}$$

and

$$f_n = \frac{N}{F^S} \tag{5.41}$$

The general instruction count of the FFBP algorithm which is governed by equation 5.26 is now modified and approximated to be:

$$
\begin{aligned}
I_C(L,S,F) &\approx \sum_{s=1}^{S} K_L[s] \times I_F[s] \times J_F[s] \times L_F[s] \times O_{MN}[s] && (5.42)\\
&\approx \sum_{s=1}^{S-1} f_l \frac{L}{f_l} F^{-s} \times f_m F^s \times f_n F^s \times F^1 \times F^{S-s} + \frac{L}{f_l} F^{-S} \times f_m F^S \times f_n F^S \times f_l F^1 \times F^{S-S} && (5.43)\\
&\approx \sum_{s=1}^{S-1} \frac{L}{F^S} F^{S-s} \times \frac{M}{F^S} F^s \times \frac{N}{F^S} F^s \times F^1 \times F^{S-s} + F^{S-S} \times M \times N \times f_l F^1 \times 1 && (5.44)\\
&\approx MN \frac{L}{F^S} F \sum_{s=1}^{S-1} +1 \times M \times N \times \frac{L}{F^S} F^1 \times 1 && (5.45)\\
&\approx \frac{LMNF(S-1)}{F^S} + \frac{LMNF}{F^S} && (5.46)\\
&\approx \frac{LMNF}{F^S}(S-1+1) && (5.47)\\
&\approx \frac{LMNF}{F^S} S && (5.48)\\
&\approx f_l MNFS && (5.49)\\
&\approx f_l MNF^1(\log_F L - \log_F f_l) && (5.50)
\end{aligned}
$$

The simplest way of expressing the instruction count is to use equation 5.48. The new modified instruction count now takes into account the scenario where $L = F^S$ and where $L \neq F^S$. In equation 5.43 it can be seen that the final stage operation count has been separated from the operation count of the intermediate processing stages. In the final stage each pixel is focused from $L_F[s] = f_l F$ range bins wheres in all other stages, $1 \leq s \leq S-1$, each new range bin value is produces by the summation of $L_F[s] = F$ previous range bins.

5.5.5 General Saving

The saving in comparison to time spent on processing an image using FFBP is now approximated by:

$$
\begin{aligned}
I_S(L,S,F) &\approx \frac{LMN}{1} \times \frac{F^S}{SMNLF} && (5.51)\\
&\approx \frac{LF^S}{SFL} && (5.52)\\
&\approx \frac{L}{l_f FS} && (5.53)
\end{aligned}
$$

The new saving of the FFBP where $L = l_f F^S$ in comparison to old FFBP saving where $L = (F_S)^S$ is approximated by:

$$
\begin{aligned}
I_S(L,S,F) &\approx \frac{F_S MNS}{1} \times \frac{F^S}{SMNLF} && (5.54)\\
&\approx \frac{F_S F^S}{LF} && (5.55)\\
&\approx \frac{F_S}{f_l F} && (5.56)
\end{aligned}
$$

$$I_D(L,S,F) \approx \frac{1}{F_S MNS} \times \frac{SMNLF}{F^s} \tag{5.57}$$

$$\approx \frac{LF}{F_S F^S} \tag{5.58}$$

$$\approx f_l \frac{F}{F_S} \tag{5.59}$$

Thus the method for implementing the practical FFBP above is only $f_l \frac{F}{F_S}$ times slower than the theoretical FFBP method explained in 4.2. For an aperture of $L = 1296$ aperture points using $S = 4$ stages of processing which are factorised by $F_s = 6$ or $F = 3$ and $f_l = 16$, focusing using the theoretical FFBP will be $\frac{L}{F_S \log_{F_S} L} = \frac{1296}{6(4)} = 54$ times faster and focusing using the practical FFBP will be $\frac{L}{l_f FS} = \frac{1296}{16(3)(4)} = 6.75$ faster than one stage filtered backprojection. Thus the practical FFBP will be $f_l \frac{F}{F_S} = \frac{1296}{3^4} 16\frac{3}{6} = 8$ times slower than the theoretical FFBP. However a saving of 6.75 is still a significant increase in speed.

5.5.6 Sub-image Processing Sequence

The loop sequence and manner in which the FFBP algorithm is recursively called, will determine how much storage resources are needed at any one time. There are however two main ways in which the loop sequence can be done .

1. The first method is to completely process one stage at a time before calling subsequent stages. This method of processing is similar to a level traversal done on a tree. The process is non-recursive and only requires loops.

2. The second is to interleave all stages thus completing the focusing of a single stage one sub-image before moving on to the next sub-image. This method of processing is similar to a preordered traversal done on a tree. The process would require recursive calls. This is the FFBP implementation used for the thesis project using five loops and recursion to reduce the instruction count of filtered backprojection.

5.5.6.1 Level Order Method

In addition to the storage needed for the input data the total amount of storage needed to produce a focused image is given by:

$$M_S \approx \sum_{s=1}^{S-1} I_F[s] \times J_F[s] \times K_L[s] \times O_{MN}[s] \tag{5.60}$$

$$\approx \sum_{s=1}^{S-1} f_l \times M \times N \tag{5.61}$$

$$\approx f_l MN(S-1) \tag{5.62}$$

where $I_F[s] \times J_F[s]$ is the amount of sub-images in each stage and $K_L[s] \times O_{MN}[s]$ is the size of the new unfocused SAR data reduced by F downrange profiles for each of the new sub-images.

This is in addition to the $M \times N$ amount of storage needed for the final image. However this implementation focuses an image to ground range. Thus the worst case scenario will be to set aside enough storage space for downrange profiles which are O long and not $O_{MN}[s]$ long. Therefore the total amount of storage required which will cater for all possible positions that the $O_{MN}[s]$ interpolated range bin values can occupy on a single downrange profile, for all intermediate stages, is approximated by:

$$
\begin{aligned}
M_S &\approx \sum_{s=1}^{S-1} I_F[s] \times J_F[s] \times K_L[s] \times O && (5.63)\\
&\approx \sum_{s=1}^{S-1} \frac{M}{F^S} F^s \times \frac{N}{F^S} F^s \times f_l F^{S-s} \times O && (5.64)\\
&\approx \sum_{s=1}^{S-1} f_l \frac{M \times N}{F^S} F^s O && (5.65)\\
&\approx f_l \frac{M \times N \times O}{F^S} \sum_{s=1}^{S-1} F^s && (5.66)\\
&\approx f_l \frac{MNO}{F^S} \frac{\left(1 - F^{S-1}\right)}{1 - F} && (5.67)
\end{aligned}
$$

It can be seen that the amount of storage increases exponentially when processing using method 1. For this reason the level ordered method is inefficient for processing on machines with low disk or memory storage capacity.

5.5.6.2 Preorder Method

The second method completes the focusing of a single first stage sub-image before moving on to the next first stage sub-image. Since this method requires the focusing of a singe pixel before moving on to focusing the next pixel, the maximum amount of storage which is needed is proportional to that of focusing a single pixel given by:

$$
\begin{aligned}
M_S &\approx \sum_{s=1}^{S-1} K_L[s] \times O && (5.68)\\
&\approx \sum_{s=1}^{S-1} L F^{-s} \times O && (5.69)\\
&\approx LO \sum_{s=1}^{S-1} F^{-s} && (5.70)\\
&\approx LO \frac{1 - F^{-(S-1)}}{1 - F^{-1}} && (5.71)
\end{aligned}
$$

It can be seen that the amount of storage needed converges to a maximum amount as more stages are used. Thus looking at equation 5.72 the maximum storage required to focus a pixel must always be more than:

$$M_S \geq LO\frac{1}{1-F^{-1}} \tag{5.72}$$

$$\geq LO\frac{F}{F-1} \tag{5.73}$$

This method is used to implement the FFBP algorithm. The above method has several advantages:

1. The amount of storage saved using the preorder method in comparison to processing using the level ordered method increases exponentially as the amount of stages needed to process increases. This can be approximated by :

$$S_M \approx \frac{LMNO}{F^{2S}}\frac{\left(1-F^{S-1}\right)}{1-F} \times \frac{F-1}{LOF} \tag{5.74}$$

$$\approx \frac{MN\left(F^{S-1}-1\right)}{F^{2S+1}} \tag{5.75}$$

   It can be seen that the saving does not take into account the number of aperture points used . Only the image size M and N as well as the factor F used and number of stages S used will determine how much of a saving there is between using the preorder method or level ordered method.

2. Since a single first stage sub-image is completely focused before the next, it is easier to implement crash recovery in the case where and unexpected event may interrupt the processing of data. It is particularly harder to implement crash recovery for the level order method since complete focusing only takes place in level S. The main problem is the vast amount of storage data which has to be kept track of in comparison to keeping track of five loop positions in stage 1. Thus at most only one first stage sub-image is lost when using the preordered processing method. Likewise it is not hard to extend crash recovery up to stage S. However keeping account of loop positions will be time consuming.

3. Because the interleaved method focuses a single first stage sub-image at a time, it easily allows the parallel processing of first stage sub-images and sub stages thereof. The level order traversal can only process sub-images in a single stage in parallel.

4. Main apertures much longer than the beamwidth can be made to process data in blocks equal to the azimuth length of the beam thus making the implementation of real time processing easier.

The sequence of loop processing used to implement the FFBP is shown in Figure 5.4.

If FFBP uses $S = 2$ stages of processing then in theory there is no need to store data since it can be used immediately in the next stage by spreading the information across the circular focusing locus. This is essentially the LBP method of processing data. Since there are only two stages of processing the information does not have to be stored when using the preordered method.

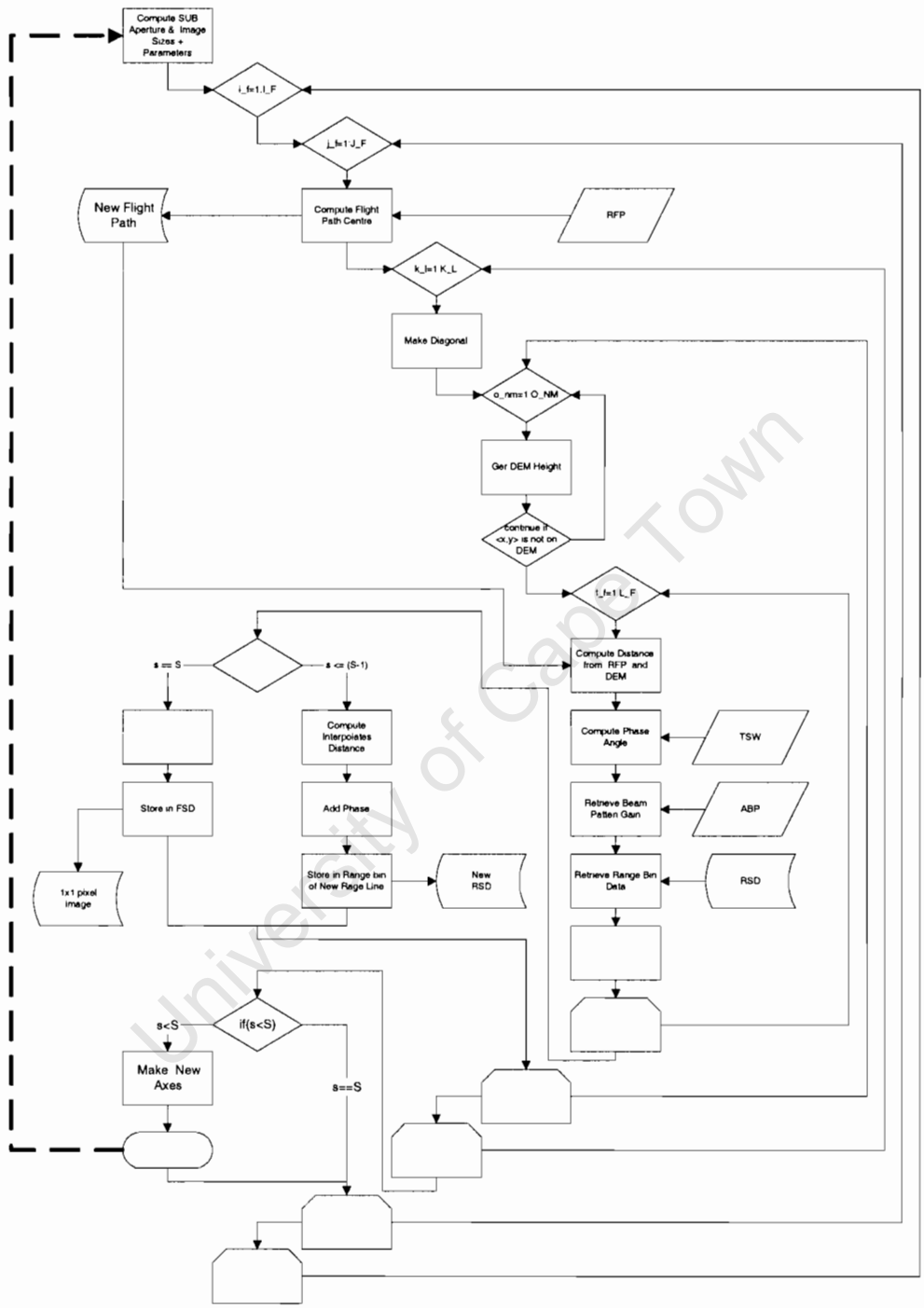


Figure 5.4: Flow diagram of fast factorised filtered backprojection

The sequence of events focusing a pixel is also briefly shown. The diagram is intended to show the five loops and recursive function calling needed to produce the focused data. The dashed line in bold shows where the recursive call is made. The first two loops are used to iterate through sub images.The recursive call is made after the completion of each sub image producing the preordered processing. The decision to focus to a pixel in stag $s = S$ or to a range bin can also be seen. The unfocused SAR data, DEM, radar flight path, transmitted signal information and antenna beam pattern all aid in producing the pixel. Output is stored in the FSD array.

5.6 Programing

The above algorithms were first all written in Matlab. Matlab was used as a prototyping language since it is much easier to debug code due to the vast amount of visual aids and easy with which data can be analysed directly from the command line .

However the implementation does not exploit the speed of the FFBP algorithm as Matlab uses an interpreter to analyse statements and execute their instruction. It was found that much more time is spent interpreting the algorithm instead of processing data making it a poor choice in terms of speed of processing for the implementation of the FFBP method described above.

SAR data set is particularly large and requires block processing or buffed processing for efficient throughput. It was found that the more information that is held in memory the more the throughput of a the CPU degrades. This is thought to be the result of the Operating System (OS) spending more time in page caching and indexing of information. A solution to the problem would be to use IO operations which access data saved on disk. This was found to be very hard to maintain for the purposes of implementing the FFBP algorithm. Thus in addition to the slow executing speed, it was also decided that Matlab was not a good choice for managing the large data set needed to implement the FFBP method of the project when working with data stored on disk

C++ on the other hand is a compiled programming language. It is significantly more efficient in IO processing that requires disk accessing. The low level programming in C++ allows one the ability to better use the full processing power of the computer. In addition C++ is an object oriented language, a feature which Matlab lacks. This would allow the array to emulate indexing by hiding the fact that the programming is indexing either disk or memory.

However C++ lacks the vast mathematical library which is present in Matlab. C++ does not handle matrices or complex data, instead only single variables or arrays. Thus the concepts of matrices and complex data had to be implemented for the running of the FFBP algorithm. Given the C++ support for abstraction and polymorphism it was possible to implement the above functions and 2D array to have a similar behaviour to that in Matlab, making porting of the FBP and FFBP algorithm easier and less error prone.

The C++ version was developed so that the practical speed of the FFBP algorithm can be compared to that of the FBP algorithm. The aim is to produce a processor that performs inversion faster than the FBP algorithm and gives images which are almost visually identical to those which are produced by normal FBP.

5.7 Data Management and Optimisation

The theoretical FFBP instruction count of $\frac{FLMNS}{F^S}$ which gives, at best, a saving of up to $\frac{L}{FS}$, for a aperture of L aperture points using S stages and a factor of F, when compared to the FBP, is not realised when focusing. The theoretical saving does not take into consideration the practical manipulation of data in the CPU.

1. SAR data size is large and must reside on disk. Indexing or storing data on this medium is considerable slower than that of memory. Even if all data is read into memory, it will be written to a swap file that will still require disk accessing. However the problem of disk processing is a problem for all algorithms which handle large data sets.

2. The FFBP algorithm splits up the aperture into many stages, making it possible to be processed on a CPU which supports a parallel architecture thus improving throughput.

This section does not discuss improving the throughput of the algorithm but addresses the practical application of disk indexing.

5.7.1 Multi-buffer Disk Arrays

Although disk IO is very efficient in C++ due to the available libraries which give a very low level ability to manipulate data on disk, it was decided to use tools, like streams which have object oriented support. Both the normal file stream functions as well as the **f-stream** classes were considered for implementation of disk IO. However **f-stream** is a class and is the better choice in an object oriented environment. Also in Linux, f-stream is just a wrapper of the file stream functions and there is a pointer to use all the file stream functionality. Using f-stream classes gives better manipulation of data through the use of the bit shift operators.

However f-stream is designed to have one small buffer which retrieves a contiguous section of file. A SAR scene is focused by retrieving and processing information found on the range locus. Due to the way data is represented, the range loci are spaced so far apart that even with the f-stream buffer it still appears as if the f-stream class indexes the disk one element at a time. This is to slow for focusing large data sets. F-stream speeds up processing only if the data on disk is accessed sequentially.

Although the FFBP algorithm processes data by splitting the problem into sub-images, the sub-images and unfocused SAR data may still be considerably large and impractical to store in memory on some machines. This could occur if the algorithm is processing a scene using two stages of processing, in which case the final sub-images may still be too large to keep in memory, without degrading data throughput.

To improve the throughput, a multi-buffer system was implemented to ensure a minimal access to disk. This buffering system was developed for the FFBP algorithm to take full advantage of the IO bandwidth.

The multi-buffer disk array encapsulates the access to different resources via a stream. In the case of this implementation the resource is a disk. The array only stores data in binary. The array accesses the disk information through a buffering system.

The array if stored as a file on disk consisting of K_A elements. Instead of having one buffer to index the whole array, the entire array is split up into I_A sectors of equal length with each sector having a separate buffer of its own. If each sector contains J_A elements the whole array is made up of $K_A = I_A \times J_A$ elements. To ensure efficient resources management each buffer must be

$$1 \leq L_A \leq J_A$$

elements. In addition there is an I_A element indexing array is used to keep the disk position of the I_A memory buffers.

The buffering system was implemented in this fashion to allow each column of a 2D set of data to have a separate buffer. Thus if an array of K_A elements is treated as a $M_A \times N_A$ 2D set of data then the number of sectors $I_A = M_A$ the number of sector elements $J_A = N_A$. Thus each column of the $K_A = M_A \times N_A$ elements array has a single buffer.

An example of the array is given in Figure 5.5.

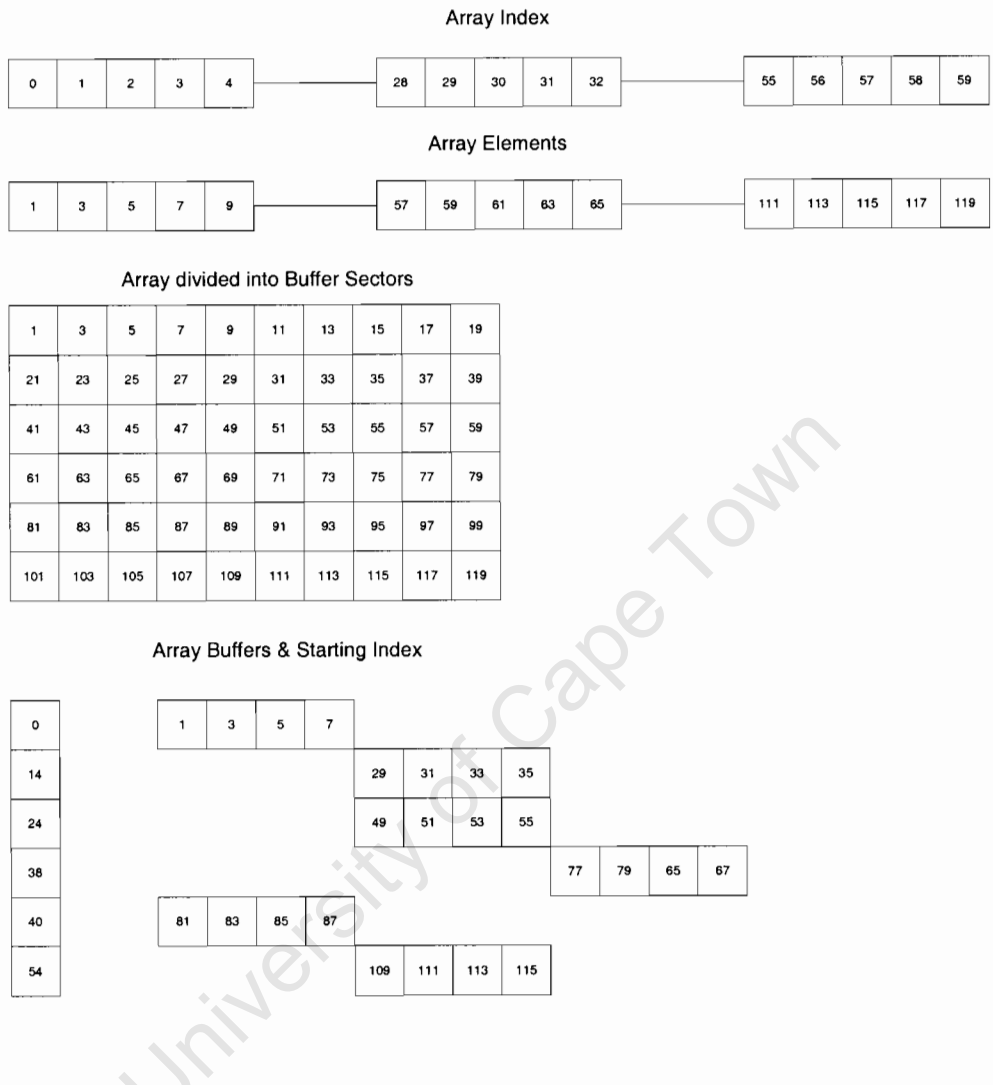


Figure 5.5: SAR stream array

The array file on disk shown with imaginary indexing axis. It also shows the array broken up into sections and the buffers in memory with the indexing buffer on the left. This array file on disk contains $K_A = 60$ elements. There are $I_A = 6$ buffers each containing $L_A = 4$ elements in each buffer. The sector length equals $J_A = \frac{K_A}{I_A} = \frac{60}{6} = 10$ elements.

Each time an element is indexed the array computes the sector in which the element must be. It then checks to see if the element resides in the current memory buffers of the appropriate row by checking the indexing array. If that element is contained in the current memory buffer, the value is returned. If it is not in the current memory buffer, the current buffer is written to disk and the block containing the information is retrieved for indexing. The indexing array is updated to reflect the start index on disk of the new current buffer. Thus the information on disk is always in sink with the information stored in memory. Figure 5.6 shows a typical input sequence and Figure 5.7 shows a typical output sequences.

In order to ensure that the individually memory buffers are not fragmented in the programs memory space, they are all stored in one contiguous memory array consisting of I_A sectors of length

L_A having a total length of :

$$O_A = L_A \times I_A \tag{5.76}$$

In this way each memory buffer is stored in a separate sector. Also the disk sector is the same as the memory buffer sector meaning, that the disk sector index i_a is the same as the memory array sector index i_a.

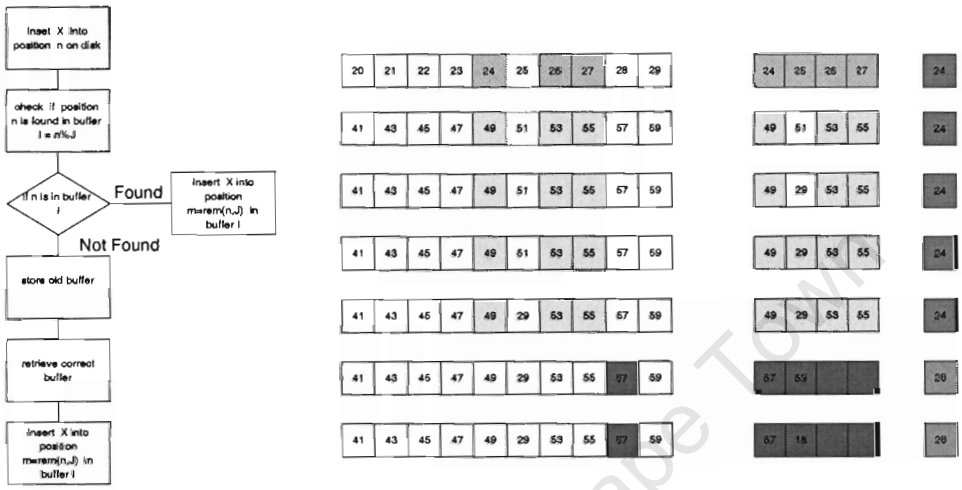


Figure 5.6: Multi buffer array disk - input

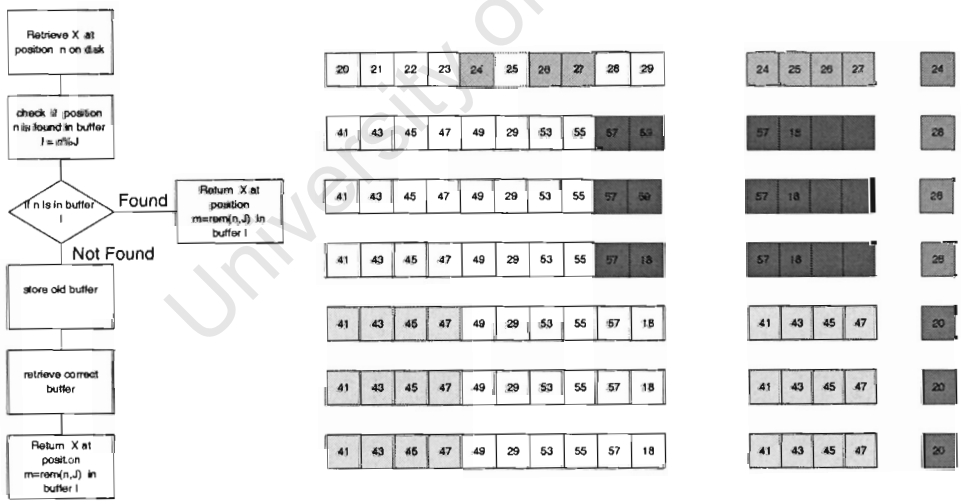


Figure 5.7: Multi buffer disk array - output

This method was chosen over using hash tables and priority queues to implement buffering or relying on the OS page changing. These buffering methods are very general and for almost all types of processing. The processing of SAR is specific and has a definite pattern to the way in which the information is index. If the loop sequence of the focusing algorithms are chosen so that pixels in the $\hat{x}$ direction are populated before the pixels in the $\hat{y}$ direction then the focusing procedure ensures that the buffers move with the range loci. Thus there is a minimum amount of disk accessing required. Thus this simple method was used which is not as time consuming and still drastically increases throughput.

Also this method is equally well suited for processing the algorithm using any of the two loop

sequences explained in Section 5.5.6. The only requirement is that there is enough disk storage available for the processing of all intermediate stages.

Thus the multi-buffer disk array provides a alternative to the method of block processing for large data sets.

Chapter 6

Testing and Results

This chapter describes the manner in which tests were conducted and reports on the results obtained from these tests. In particular, interest is in the practical (real) speed of focusing by FFBP processing in comparison to that of the FBP processing. The visual differences of the two focusing methods are also highlighted. In order to show this the images were enhanced by clipping off all information above the $12dB$ mark to see the defects which the FFBP algorithm brings due to the approximations made during focusing, by taking the scenes focused by FBP as the standard.

The chapter starts by giving a description of the scene which was used to carry out the testing. Thereafter the main testing of the FBP and FFBP algorithms was done. A real scene processed by both FBP and FFBP is also presented and finally the testing of the multi-buffer systems is discussed.

6.1 Scene Description

The description of the input data used to simulate the unfocused SAR scene is given below. The setup information can be found in Tables 6.1 to 6.2 and diagrams of the 2D data sets are shown in Figure 6.1.

The test scene consists of a radar flight path with a maximum deviation of 36 meters from the y axis and nominal 6.1 (a), a DEM with a 48 meters difference between the highest and lowest point in Figure 6.1 (b), a beam width which is 90^0in azimuth and 30^0 in swath Figure 6.1 (c) and a ground reflection map with point targets spread out by being spaced at 24 meter intervals in both x and y direction.

Description	Dimensions	Value	Unit
Wave speed	c	$3e8$	(ms^{-1})
Voltage	V_t	1	(V)
Centre frequency	f_c	466	(MHz)
Bandwidth	Δf	300	(MHz)

Table 6.1: Information of transmitted signal waveform, Figure 6.1 (a), simulation scene (A)

Table 6.6 gives the setup information used to define the 2D array which was used for the simulation of the unfocused range compressed SAR data. Both the simulations and focusing of data in Figure 6.2 were done with the aid of the radar flight path , DEM and antenna beam pattern as described in the flow diagrams of Figures 5.2 and 5.3.

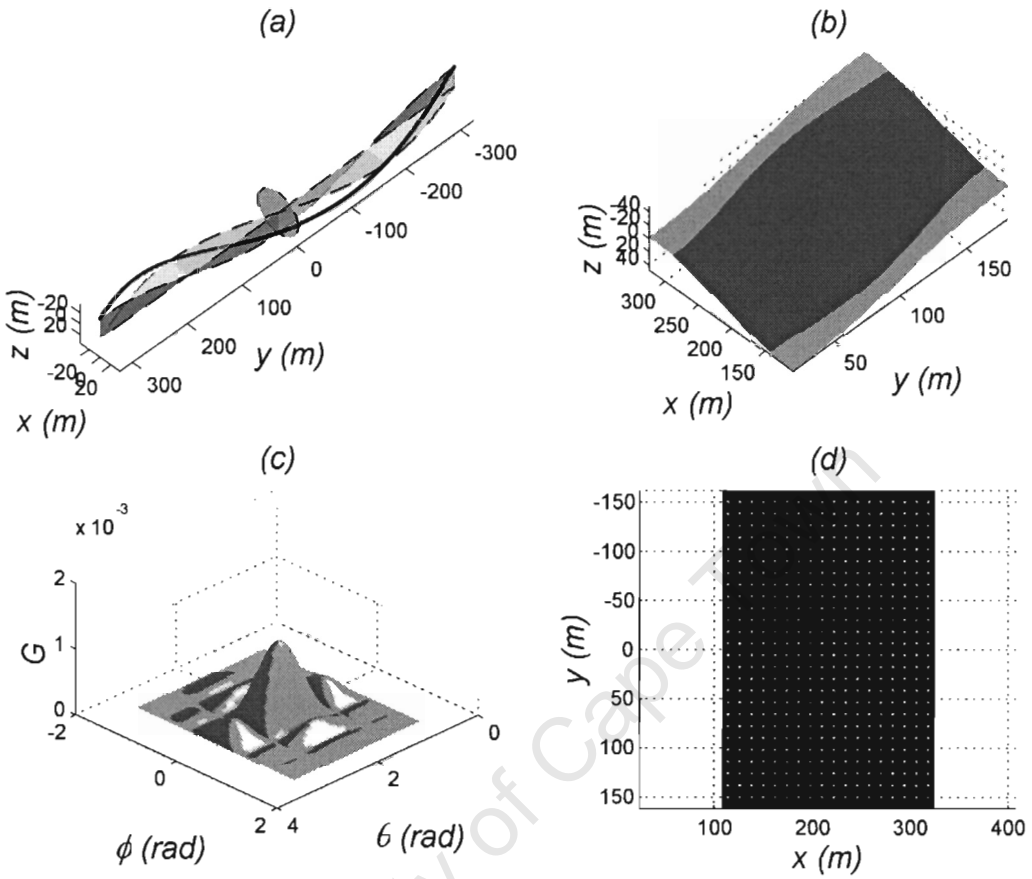


Figure 6.1: Graphical representation of 2D input data
(a) shows the radar flight path which corresponds to Table 6.2, (b) shows the DEM which corresponds to Table 6.3, 6.1 (c) shows the antenna beam pattern which corresponds to Table 6.4 and (d) shows the RCS which corresponds to Table 6.5.

Description	Dimensions	Value	Unit
RFP Resolution (Average)	dl	0.5	$[m]$
RFP Vector Length	L	1296	$[l]$
Squint Angle	θ_{Squint}	$0(0^0)$	$[rad\,(^0)]$
Depression Angle	$\phi_{Depression}$	$0(36.8^0)$	$[rad\,(^0)]$

Table 6.2: Information of Radar flight path, Figure 6.1 (b), simulation scene (A)

Description	Dimensions	Value	Unit
DEM Centre	$(X_I \times Y_J)$	216×162	$(m \times m)$
DEM Dimensions	$(\Delta x \times \Delta y)$	432×648	$(m \times m)$
DEM Resolution	$(dy \times dx)$	0.5×0.5	$(m \times m)$
DEM Image Size	$[I_X \times J_Y]$	432×648	$[i_x \times j_j]$
DEM Centre Height	Z_c	162	(m)

Table 6.3: Parameters for DEM parameters of Figure 6.1 (c), simulation scene (A)

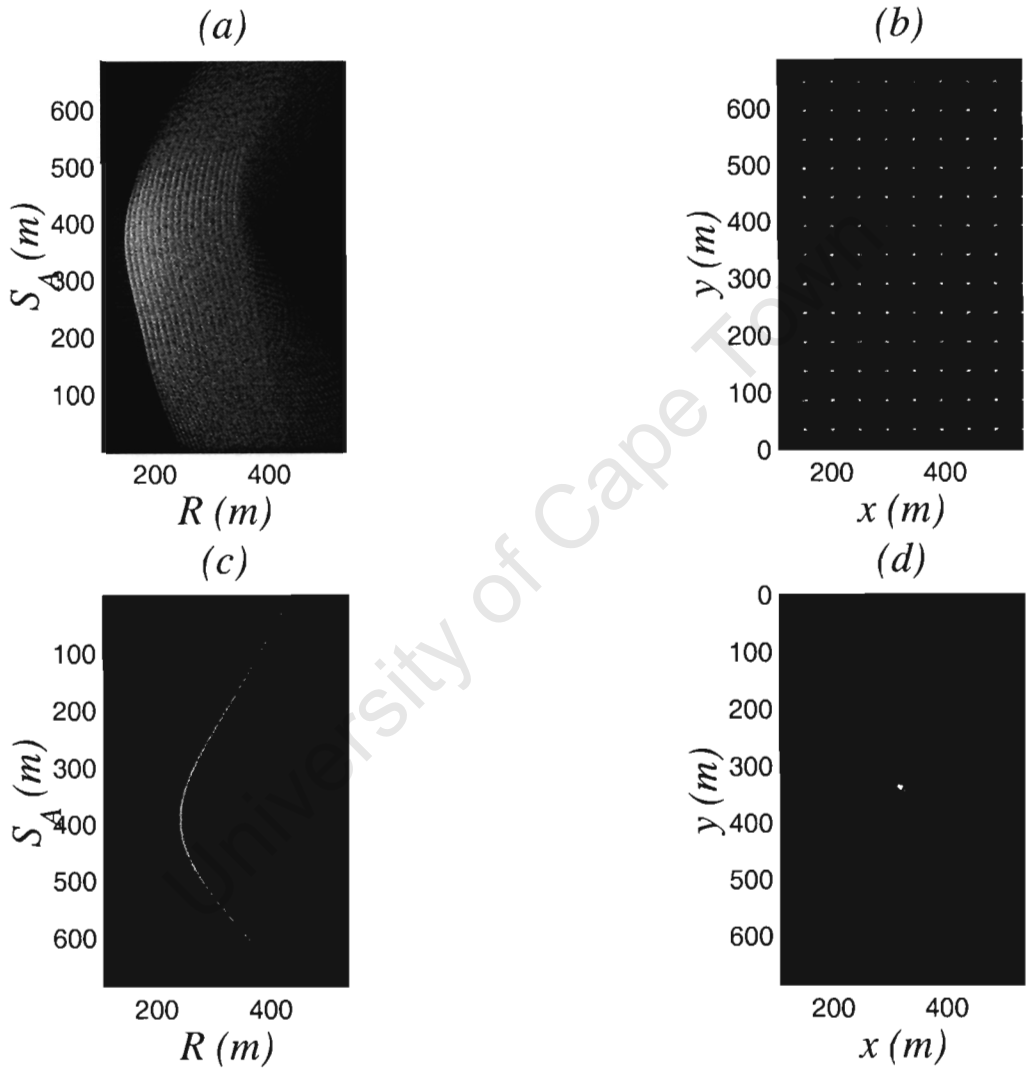


Figure 6.2: Simulation of unfocused SAR data, simulation scene (A)
(a) show the simulated unfocused SAR data for simulation scene (A) and (b) shows the scene focused using FBP. Figure 6.2 (c) show the simulated unfocused SAR data of a single point target and (d) shows the focusing of the point target using FBP. The unfocused SAR data 2D array in (a) and (c) corresponds to Table 6.6

Description	Dimensions	Value	Unit
ABP Centre	$(\Theta_A \times \Phi_E)$	0×0	$(rad \times rad)$
ABP Dimensions	$(\Delta\theta_A \times \Delta\phi_E)$	$2\pi \times \pi$	$(rad \times rad)$
ABP Resolution	$(d\theta_A \times d\phi_E)$	$\frac{\pi}{1024.5} \times \frac{\pi}{65}$	$(rad \times rad)$
ABP Image Size	$[I_\Theta \times J_\Phi]$	2049×65	$[i_\theta \times j_\phi]$

Table 6.4: Parameters for antenna beam pattern footprint, of Figure 6.1 (c), simulation scene (A)

Description	Value	Unit	Dimensions
GRM Centre	$(X_\sigma \times Y_\sigma)$	0×216	$(m \times m)$
GRM Dimensions	$(\Delta x_i \times \Delta y_j)$	216×324	$(m \times m)$
GRM Resolution	$(dx_i \times dy_j)$	0.5×0.5	$(m \times m)$
GRM Image Size	$[I \times J]$	432×648	$[i \times j]$

Table 6.5: Parameters for ground reflectivity map footprint of Figure 6.1 (d), simulation scene (A)

A few more useful variables describing the scene are given in Table 6.7. This information is not needed for simulation or focusing but helps in giving a better visual description of the scene.

The maximum deviation of the radar flight path in Figure 6.1 (a) from the y axis is 36 meters and the difference between the highest and lowest point on the DEM in 6.1 (b) is 48 meters.

6.2 SAR Focusing Results

The backprojection tests which follow will perform SAR inversion on the unfocused range compressed SAR data illustrated in Figure 6.2 (a). Section 6.2.1 deals with the necessity of using a radar flight path, DEM and antenna beam pattern for the inversion of unfocused SAR data acquired from a nonlinear flight path and targets located on an uneven ground using a radar which produces a wide beam width. The section thereafter deals with the efficiency of the FFBP algorithm in reducing the time of processing by speeding up the backprojection process. This is done by comparing the time taken to process a scene using FFBP to the time taken to process the same scene using FBP. Section 6.2.2 will also comment on the quality of a image processed by FFBP in comparison to an image processed by FBP.

6.2.1 FBP Testing

The results of the backprojection focusing are given below. Table 6.8 gives the information describing the 2D ground range array on the $x - y$ plane. Figure 6.3 shows the processing of the same SAR scenes all done in a different manner.

Figure 6.3 clearly shows how the different sets of 2D input data contribute to producing an accurately focused scene. As can be seen Figure 6.3 (a) gives the most accurately focused image of the

Description	Value	Unit	Dimensions
USD Start	$(T_I \times R_I)$	0×108	$(s \times m)$
USD Dimensions	$(\Delta t_A \times \Delta R)$	648×432	$(s \times m)$
USD Resolution	$(dt_A \times dR)$	0.5×0.25	$(s \times m)$
USD Image Size	$[L \times O]$	1296×1728	$[l \times o]$

Table 6.6: Parameters for simulated unfocused SAR data, simulation scene (A)

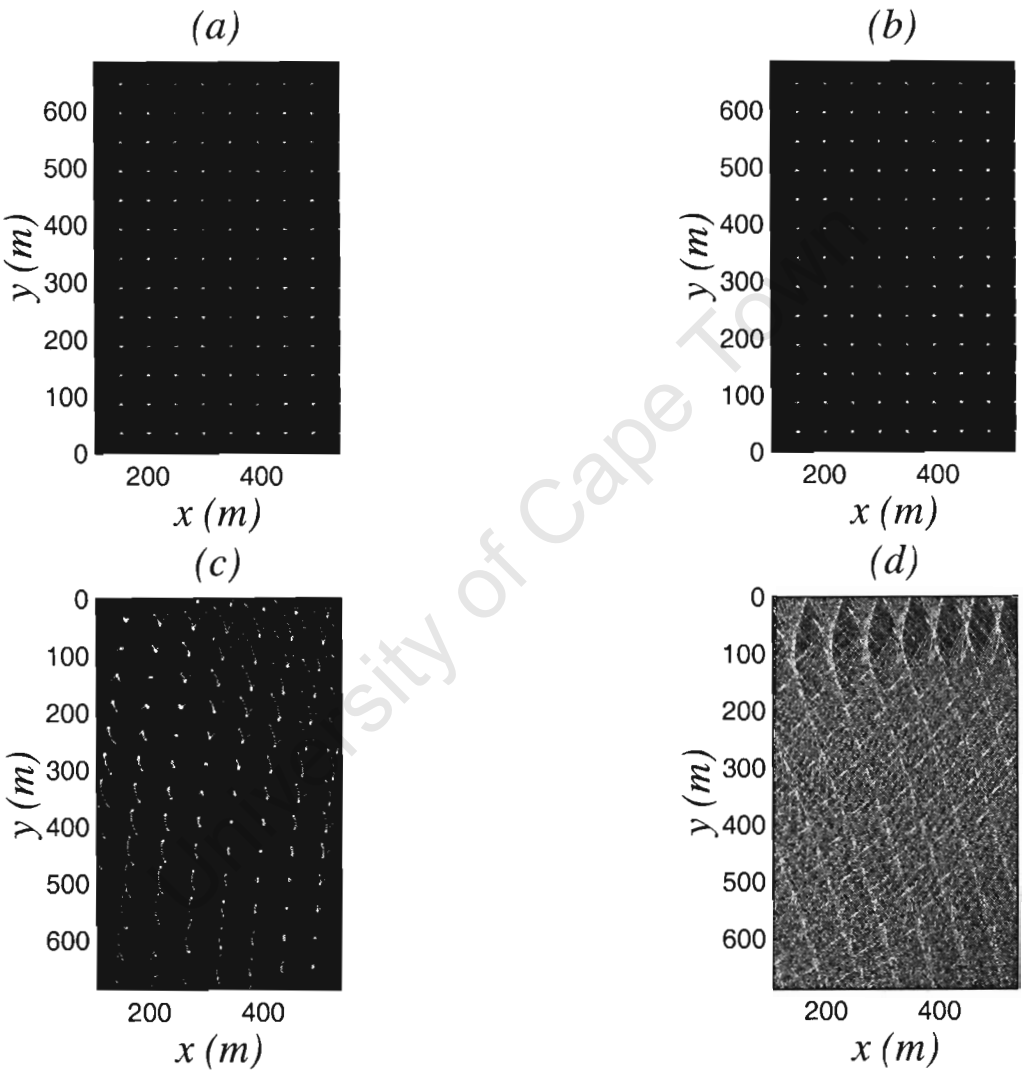


Figure 6.3: FBP focusing scenarios,simulation scene (A), focused scene (I)
In (a) the scene is processed with the aid of the antenna beam pattern, DEM and radar flight path . Then in (b) the scene is processed with the aid of a the DEM and radar flight path, in (c) scene is processed with the aid of a radar flight path only and (d) assumes a linear aperture and and processes the scene without radar flight path, DEM and antenna beam pattern.

Description	Value	Unit
Flight Path Length	648	(m)
Flight Path Time	10.8	(s)
PRI	8.33	(ms)
Flight Path Points	1296	$[l]$
Average Flight Path Resolution	0.5	(m)
Average Speed	216	(ms^{-1})
Swath Beam Width	90	$(^0)$
Elevation Beam Width	30	$(^0)$

Table 6.7: Extra information, simulation scene (A)

Description	Value	Unit	Dimensions
FSD Centre	$(X_C \times Y_C)$	216.5×0.5	$(m \times m)$
FSD Dimensions	$(\Delta x_m \times \Delta y_n)$	162×108	$(m \times m)$
FSD resolution	$(dx_m \times dy_n)$	0.25×0.25	$(m \times m)$
FSD Image Size	$[M \times N]$	432×648	$[m \times n]$

Table 6.8: Parameters for focused SAR data ground range image grid, focused scene (A)

four diagrams in Figure 6.3. It can also be seen in Figure 6.3 (b) that processing the scene without the aid of the antenna beam pattern does not drastically degrade the quality of the image. However the uneven ground used to simulate the scene requires a DEM in order to properly retrieve the focused scene as can be seen in Figure 6.3 (c). Figure 6.3 (d) shows the main advantage that backprojection has over other methods which assume apertures to be linear. The processing of a scene without DEM ,radar flight path and antenna beam pattern in Figure 6.3 (d) cannot focus the scene. When focusing from a linear aperture, instead of taking the real flight path into account, it is almost impossible to reconstruct the scene. This particular simulation in Section 6.1 used for testing incorporated a huge deviations much greater than the resolution due to band width. This deviation is much greater that $\frac{\delta R}{2}$ making it impossible to focus the image using FBP without a radar flight path.

6.2.2 FFBP Testing

The main emphasis of this section deals with the ratio of the time spent processing a scene using FBP in comparison to FFBP. It has been mentioned in previous sections that the time saving ratio I_S of equation 5.51 does not take into account the real time of processing. Since the real time saving ratio will always be less than the theoretically quoted value, it would be interesting to see how processing a scene with different factors F and stages S will affect the ratio of time saved by processing an image using FFBP instead of processing the scene using FBP. Thus in addition to testing the accuracy of the time saving ratio, the combinations of factors F and stages S which give the least deviation from the theoretical time saved are also investigated .

To do this the testing was split up into four batches each containing three FFBP tests. Each batch of tests has a different relationship between the factor F and stage S. These batches were focused as follows:

1. The scenes were processed using factors F and stage S, strictly obeying equation 4.8 where $L = F^S$.

2. The scenes were processed by keeping the factor F constant, while the number of stages S were varied .

3. The scenes were processed by setting $S = S_F$ where $L = f_l F^S$ and $f_l \neq 1$.

4. The scenes were processed by keeping S constant and allowing $L \approx F^S$.

The spatial dimensions of the 2D array used for the tests in this section tests are much smaller than that used in the test for section 6.2.1[1]. Table 6.9 gives the setup information to produce the 2D ground range array. The array contains the same amount of pixels, however the resolution of the ground range image is three time higher in both x and y direction. This was done to accurately compare the focusing of the point spread distributions done with FBP to that done with FFBP.

Description	Value	Unit	Dimensions
FSD Centre	$[X_C \times Y_C]$	216.5×0.5	$[m \times m]$
FSD Dimensions	$[\Delta x_m \times \Delta y_n]$	54×36	$[m \times m]$
FSD resolution	$[dx_m \times dy_n]$	$\frac{1}{12} \times \frac{1}{12}$	$[m \times m]$
FSD Image Size	$[M \times N]$	432×648	$[m \times n]$

Table 6.9: Parameters for focused SAR data image grid , simulation scene (A), focused scene (II)

Although each batch contains only three tests there are four images in all the figures presented. Image (a) is alway the scene processed by FBP and the other three images are processed by FFBP. In addition to the focused results there are tables giving factorisation information on the the images. The tables are arranged from top to bottom so that they correspond to the alphabetic labelling of the their corresponding images. The rows of each table are partitioned into sections which will be referred to here as levels. Each table is arranged into 7 levels. The columns are divided into three sections. A description column, a left hand side and right hand side consisting of three columns each.

- The 1^{st} level shows the input dimensions on the left hand side and the factorisation information on the right hand side.

- The 2^{nd} level gives information on various constants. The left hand side gives constants used in the factorisation process and the right side shows the constants which give guide lines to the maximum number of stages practical for processing, using a factor F.

- The left hand side of the 3^{rd} level gives the pixel dimensions of the sub-images and sub-apertures and the right hand side gives the total number of sub-images and sub-apertures in stage $s = 1$.

- The 4^{th} level has the same format as the 3^{rd} level but for the final stage $s = S$ of processing. The only difference between the 4^{th} and 3^{rd} levels is that the 4^{th} level shows the number of sub-images, sub-apertures and corresponding pixel dimensions in which the final stage S starts off with, whereas the 3^{rd} level shows the number of sub-images, sub-aperture and corresponding pixel dimensions which is produced in stage $s = 1$.

[1]The spacial dimensions do not affect the speed of image focusing. The instruction count is dependent on the $M \times N$ pixel that have to be produced in the final image. Section 6.2.1 and 6.2.2 both have the same amount of pixel in the final image but focus scenes with different spacial dimensions.

- The 5^{th} level gives the instruction count for stage $s = 1$ on the left hand side and the instruction count for stage $s = S$ on the right hand side based on the information in the previous two levels.
- The 6^{th} level gives various instruction count statistics and
- The 7^{th} level gives the instruction count and time taken to process the images on the left hand side. The first line shows the instruction count and time it takes to process an image using for FBP. Line 2 shows the theoretical instruction count and time taken for FFBP processing using factor F and stages S. The final line shows the real instruction count and time it takes to process an image. The right hand side shows the difference(deviation) from the theoretical time saving I_S.

6.2.2.1 Test Results: Batch 1

The results of the first test are shown in Figure 6.4 and Figure 6.5 gives the factorisation information and time taken to complete the inversion process. The test are done using $S = \{1, 2, 3, 4\}$ stages and since $L = F^S$ the focused images are representative of FFBP SAR inversion using factors $F = \left\{1296, 36, \sqrt[3]{1296}, 6\right\}$. Figure (a) shows the processing of the scene processed by FBP, therefore $F = 1296$ and $S = 1$. Figure (b) ,(c) and (d) shows the scenes processed by FFBP using Factors and stages $F = 36$ & $S = 2$, $F = \sqrt[3]{1296}$ & $S = 3$ and $F = 6$ & $S = 4$ respectively.

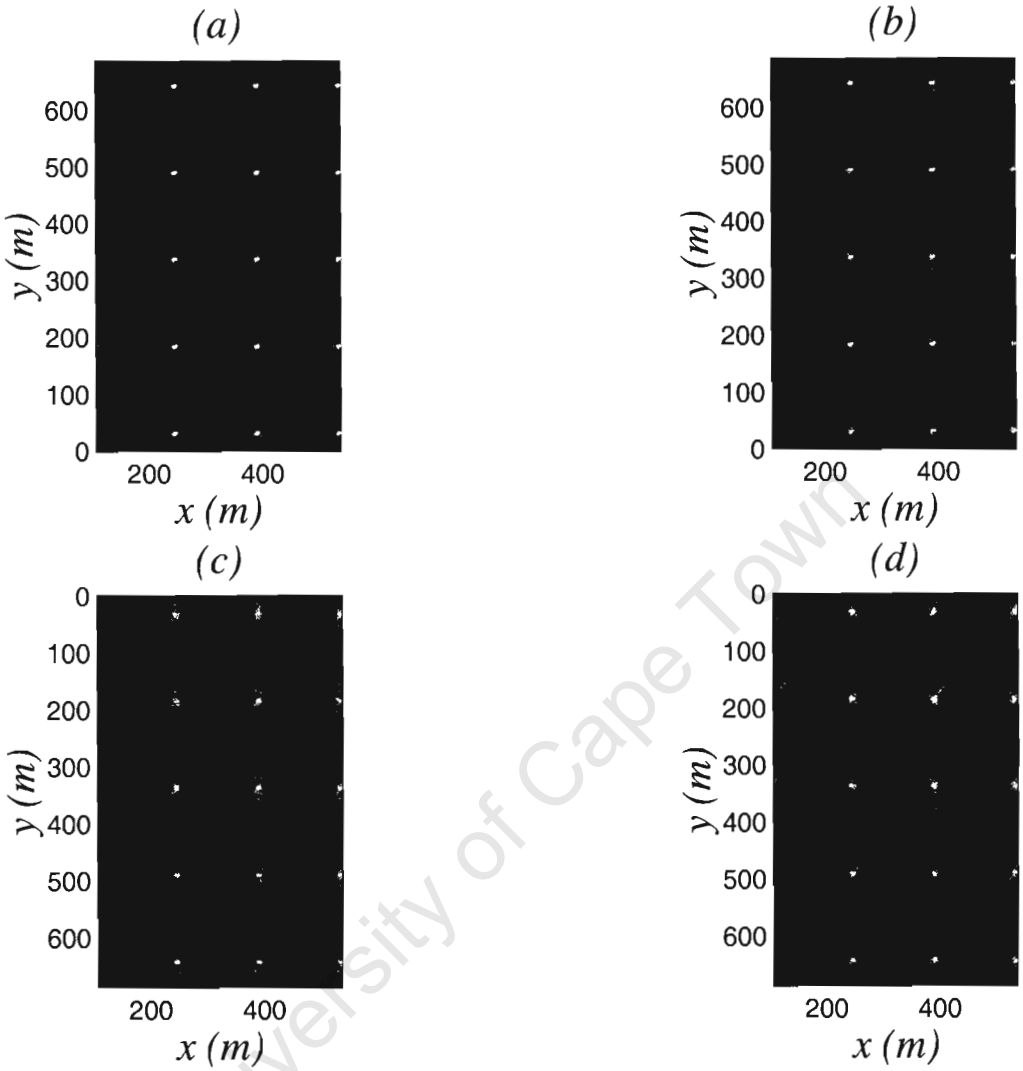
At close inspection it can clearly be seen how the images degrades from Figure 6.5 (a) to (d) as more stages are used to process an image. There is more dispersion of the point spread distribution, when processing images using more stages. Also when looking at the time it takes to process an image it can be seen that the focusing is faster when using more stages, however the ratio of time saved deviated more from the predicted value as more stages are used.

6.2.2.2 Test Results: Batch 2

The second batch of test were preformed by keeping the factor $F = 6$ and by varying the amount of processing stages S used to complete the focusing of an image using FFBP. Figure 6.6 (a) shows the result of one stage FBP processing and Figures 6.6 (b) ,(c) and (d) shows the focusing of the images processed by FFBP using $S = \{2, 3, 4\}$ stages. Figure 6.7 gives the factorisation information and time of focusing for the respective scenes in Figure 6.6. Again it can be seen that the images degrade slightly as more stages are used due to the approximations made by the FFBP algorithm. Thus the more stages which are used, the more the accumulated approximations will degrade the image. It can also be seen since F is constant, that the difference between the real time and the theoretical time increases the more L_S deviates from L. This is due to practical data management which will be discussed in Section 6.2.2.5.

6.2.2.3 Test Results: 3

Test batch 3 was done by setting the stages $S = S_F$ for the given factors $F = \{3, 4, 2\}$. Figure 6.8 shows the results with corresponding table in Figure 6.9. Figure 6.8 (a) represents the focused scene processed by FBP and Figures 6.8 (b), (c) and (d) shows the results of the focused scene processed by FFBP using factors $F = \{3, 4, 2\}$ with stages $S = \{5, 4, 8\}$ respectively.



(a)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	1296	1	1296
	count	time	saving			
LMN	362797056	1946.209	1			
f_1 SFMN (THEORY)	362797056	1946.209	1	Difference		
f_1 SFMN (REAL)	362797056	1946.209	1	1		

(b)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	36	2	1296
	count	time	saving			
LMN	362797056	1946.209	1			
f_1 SFMN (THEORY)	20155392	108.1227222	18	Difference		
f_1 SFMN (REAL)	20155392	114.064	17.06242986	1.054949101		

(c)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	10.90272356	3	1296
	count	time	saving			
LMN	362797056	1946.209	1			
f_1 SFMN (THEORY)	9156194.465	49.11800628	39.62312699	Difference		
f_1 SFMN (REAL)	8235216	57.343	33.93978341	1.167453737		

(d)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	6	4	1296
	count	time	saving			
LMN	362797056	1946.209	1			
f_1 SFMN (THEORY)	6718464	36.04090741	54	Difference		
f_1 SFMN (REAL)	6718464	57.002	34.14281955	1.5815917		

Figure 6.4: FFBP Test Results, simulation scene (A), focused scene (II), Batch 1

(a)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	1296	1	1296
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	1	0.333333333	0.5	1	10	1
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	1296	1	1	432	648	1
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	1296	1296	1296	0.333333333	0.5	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	1296	279936	362797056	2176782336	0.166666667	362797056
Instruction Count	LMN	f_l SFMN	f_l FMN	SFNM	FNM	eln(L)NM
	362797056	362797056	362797056	362797056	362797056	5453721.221
	count	time	saving			
LMN	362797056	1946.209	1			
f_l SFMN (THEORY)	362797056	1946.209	1	Difference		
f_l SFMN (REAL)	362797056	1946.209	1	1		

(b)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	36	2	1296
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	1	0.333333333	0.5	2	10	2
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	36	36	36	12	18	36
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	36	36	36	12	18	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	46656	216	10077696	46656	216	10077696
Instruction Count	LMN	f_l SFMN	f_l FMN	SFNM	FNM	eln(L)NM
	362797056	20155392	10077696	20155392	10077696	5453721.221
	count	time	saving			
LMN	362797056	1946.209	1			
f_l SFMN (THEORY)	20155392	108.1227222	18	Difference		
f_l SFMN (REAL)	20155392	114.064	17.06242986	1.054949391		

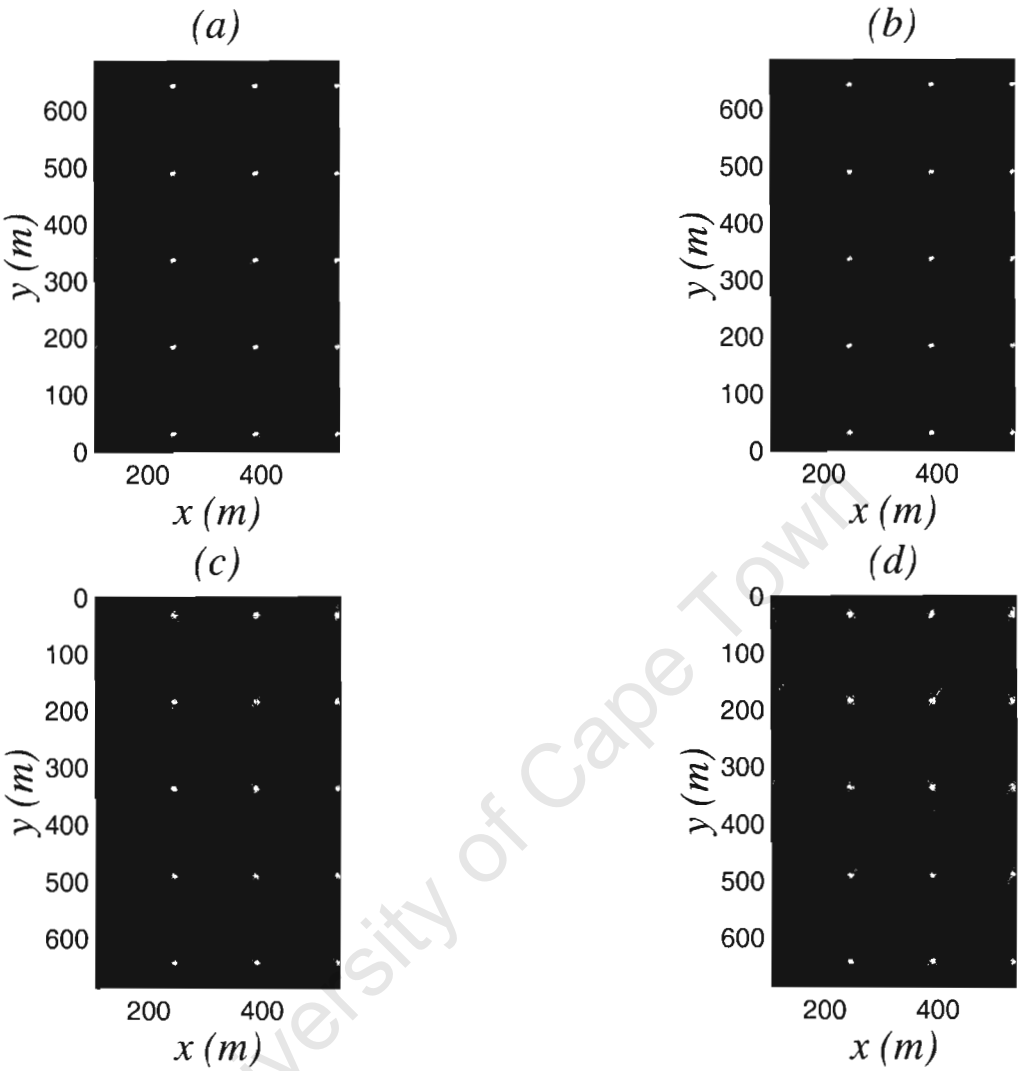
(c)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	10.90272356	3	1296
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	1	0.333333333	0.5	3	10	3
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	10.90272356	118.869381	118.869381	3.634241186	5.451361778	118.869381
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	10.90272356	10.90272356	10.90272356	39.62312699	59.43469048	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	154054.7177	19.81156349	3052064.822	1296	2354.988288	3052064.822
Instruction Count	LMN	f_l SFMN	f_l FMN	SFNM	FNM	eln(L)NM
	362797056	9156194.465	3052064.822	9156194.465	3052064.822	5453721.221
	count	time	saving			
LMN	362797056	1946.209	1			
f_l SFMN (THEORY)	9156194.465	49.11800628	39.62312699	Difference		
f_l SFMN (REAL)	8235216	57.343	33.93978341	1.167453737		

(d)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	6	4	1296
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	1	0.333333333	0.5	4	10	4
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	6	216	216	2	3	216
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	6	6	6	72	108	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	279936	6	1679616	216	7776	1679616
Instruction Count	LMN	f_l SFMN	f_l FMN	SFNM	FNM	eln(L)NM
	362797056	6718464	1679616	6718464	1679616	5453721.221
	count	time	saving			
LMN	362797056	1946.209	1			
f_l SFMN (THEORY)	6718464	36.04090741	54	Difference		
f_l SFMN (REAL)	6718464	57.002	34.14281955	1.5815917		

Figure 6.5: Information of FFBP Test Results, simulation scene (A), focused scene (II), Batch 1



(a)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	1296	1	1296
	count	time	saving			
LMN	362797056	1946.209	1			
f_1 SFMN (THEORY)	362797056	1946.209	1	Difference		
f_1 SFMN (REAL)	362797056	1946.209	1	1		

(b)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	6	2	36
	count	time	saving			
LMN	362797056	1946.209	1			
f_1 SFMN (THEORY)	120932352	648.7363333	3	Difference		
f_1 SFMN (REAL)	120932352	1387.095	1.403082702	2.138149089		

(c)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	6	3	216
	count	time	saving			
LMN	362797056	1946.209	1			
f_1 SFMN (THEORY)	30233088	162.1840833	12	Difference		
f_1 SFMN (REAL)	30233088	286.182	6.80059892	1.764550467		

(d)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	6	4	1296
	count	time	saving			
LMN	362797056	1946.209	1			
f_1 SFMN (THEORY)	6718464	36.04090741	54	Difference		
f_1 SFMN (REAL)	6718464	57.002	34.14281955	1.5815917		

Figure 6.6: FFBP Test Results, simulation scene (A), focused scene (II), Batch 2

(a)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	1296	1	1296
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	1	0.333333333	0.5	1	10	1
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	1296	1	1	432	648	1
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	1296	1296	1296	0.333333333	0.5	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	1296	279936	362797056	2176782336	0.166666667	362797056
Instruction Count	LMN	f_l SFMN	f_l FMN	SFNM	FNM	eln(L)NM
	362797056	362797056	362797056	362797056	362797056	5453721.221
	count	time	saving			
LMN	362797056	1946.209	1			
f_l SFMN (THEORY)	362797056	1946.209	1	Difference		
f_l SFMN (REAL)	362797056	1946.209	1	1		

(b)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	6	2	36
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	36	12	18	4	10	4
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	6	6	6	72	108	216
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	216	6	6	72	108	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	7776	7776	60466176	7776	7776	60466176
Instruction Count	LMN	f_l SFMN	f_l FMN	SFNM	FNM	eln(L)NM
	362797056	120932352	60466176	3359232	1679616	5453721.221
	count	time	saving			
LMN	362797056	1946.209	1			
f_l SFMN (THEORY)	120932352	648.7363333	3	Difference		
f_l SFMN (REAL)	120932352	1387.095	1.403082702	2.138149089		

(c)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	6	3	216
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	6	2	3	4	10	4
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	6	36	36	12	18	216
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	36	6	6	72	108	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	46656	216	10077696	1296	7776	10077696
Instruction Count	LMN	f_l SFMN	f_l FMN	SFNM	FNM	eln(L)NM
	362797056	30233088	10077696	5038848	1679616	5453721.221
	count	time	saving			
LMN	362797056	1946.209	1			
f_l SFMN (THEORY)	30233088	162.1840833	12	Difference		
f_l SFMN (REAL)	30233088	286.182	6.80059892	1.764550467		

(d)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	6	4	1296
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	1	0.333333333	0.5	4	10	4
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	6	216	216	2	3	216
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	6	6	6	72	108	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	279936	6	1679616	216	7776	1679616
Instruction Count	LMN	f_l SFMN	f_l FMN	SFNM	FNM	eln(L)NM
	362797056	6718464	1679616	6718464	1679616	5453721.221
	count	time	saving			
LMN	362797056	1946.209	1			
f_l SFMN (THEORY)	6718464	36.04090741	54	Difference		
f_l SFMN (REAL)	6718464	57.002	34.14281955	1.5815917		

Figure 6.7: Information of FFBP Test Results, simulation scene (A), focused scene (II), Batch 2

In particular interest is Figure 6.8 (d) which uses a factor $F = 2$ and $S = 8$ stages to complete processing. Firstly the dispersion of the point spread distribution is much worse because of the many stages being used. Also the deviation of the theoretically estimated time and the real time taken is much greater than the theoretically predicted value. The deviation for 6.8 (b) and (c) is however still tolerable but the estimated time to focus an image deviates much more from the theoretically predicted value than in the previous test batches.

6.2.2.4 Test Results: Batch 4

In the previous test it was noted that there is a great difference between the real and theoretical estimated times when $L \neq F^S$. In order to test if the best time saving occurs when $L = F^S$ the test here are done by setting $L \approx F^S$. This is done by using $S = 3$ stages and by using values $F \approx \sqrt[3]{1296}$. As in all tests batches 1, 2 and 3 Figure 6.10 shows the results and Figure 6.11 gives a table displaying the factorisation and timing information. Figure 6.10 (a) is the processed by FBP and 6.10 (b), (c) and (d) are processed by FFBP using factors $F = \{12, 11, 10\}$ respectively. As can be seen the difference between the real and theoretical saving is not much worse than in test batch 1.

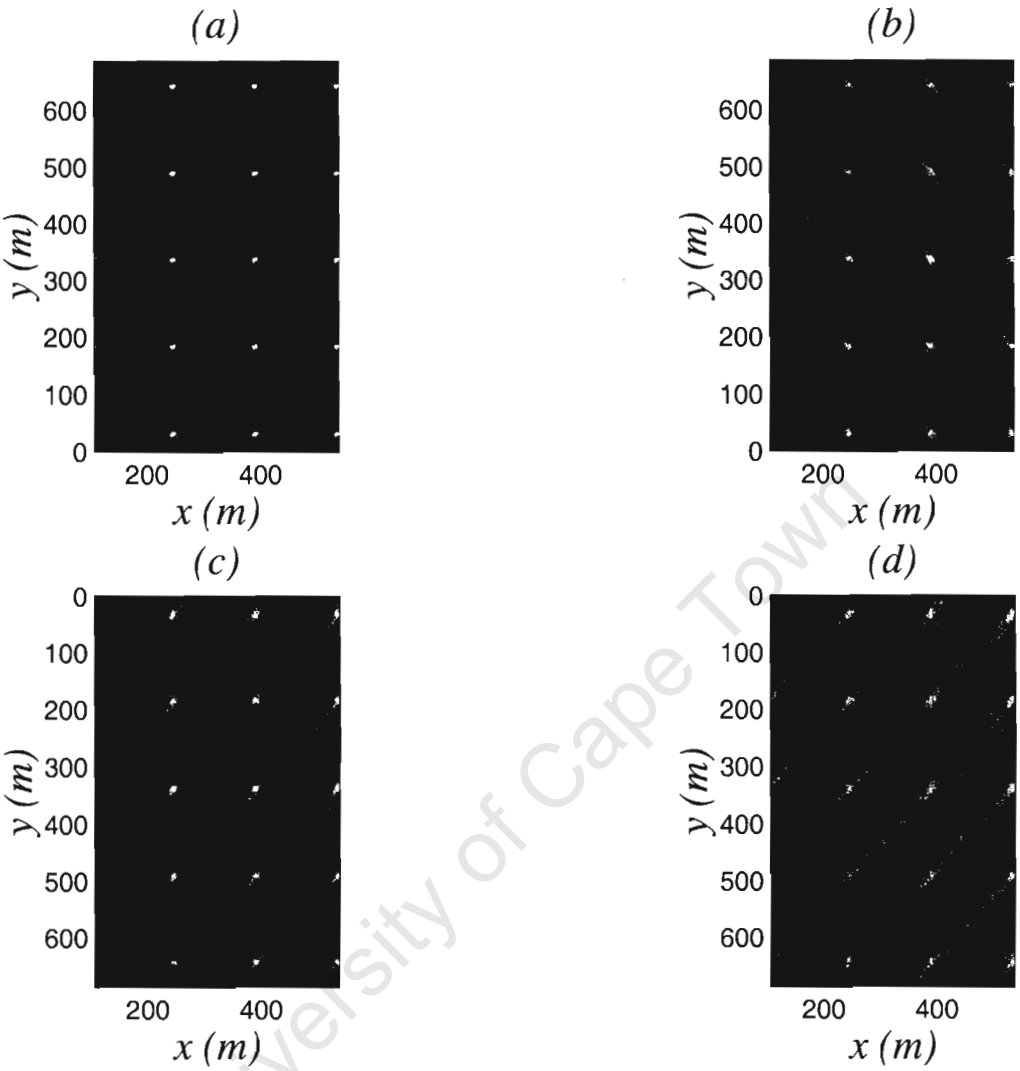
6.2.2.5 Discussion of Results

The above batch of tests shows that up to four stages of processing still produces a reasonable focused image. However it can be seen in test batch 2 (b) & (c) and test batch 3 (b), (c) & (d) that there is a huge deviation in time from the theoretically estimated and real times.

It has been observed that this occurs when $L \neq F^S$. The extra time in processing is most likely caused by data management and the recursive function calling.

It can also be seen that if $L = F^S$ then the number of aperture points progressively deceases through the S stages of processing in proportion to the exponential increase of sub-imaged. If $L \neq F^S$ it can be seen that the final stages require many more downrange profiles to focus a pixel of a sub-image, increasing the time to process an image. This can be seen in test batch 2. In the final stage of processing all scenes are focused by using 7776, 6×6 pixel sub-images. However when using $S = 2$ stages each pixel is focused by 216 range bin values, when using $S = 3$ stages each pixel is focused by 36 range bin values and when using $S = 4$ stages each pixel is focused by 6 range bin values. Therefore in stage S there is an increase of 6 times the instruction count using different number of stages $S = \{4, 3, 2\}$ to focus an image. Since all stages have an equal number of instructions there is an overall 6 times increase in the number of instructions required to complete the focusing process. This is however only the theoretical increase. The real difference will inevitably be much more pronounced when focusing an image using $S = 2$ stages than focusing using $S = 4$ stages because of data management and recursive function calling. Carrying 216 downrange profiles to the final stage for each of the 7776 images takes much more storage space and time to manage than carrying 6 downrange profiles to the final stage for each of the 7776 images .

In test batch 4 where $L \approx F^S$ the effect of data management is not so pronounced. However in test batch 3 it can be seen using $S > 4$ stages drastically slows down the processing due to recursive function calling. Even when using a factor with $F = 2$ which should not result in requiring a lot more storage resources for intermediate processing, the algorithm still has a huge difference to the



(a)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	1296	1	1296
	count	time	saving			
LMN	362797056	1946.209	1			
f_1SFMN (THEORY)	362797056	1946.209	1	Difference		
f_1SFMN (REAL)	362797056	1946.209	1	1		

(b)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	3	5	243
	count	time	saving			
LMN	362797056	1946.209	1			
f_1SFMN (THEORY)	22394880	120.136358	16.2	Difference		
f_1SFMN (REAL)	22394880	440.844	4.41473401	3.669530251		

(c)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	4	4	256
	count	time	saving			
LMN	362797056	1946.209	1			
f_1SFMN (THEORY)	22674816	121.6380625	16	Difference		
f_1SFMN (REAL)	22605696	383.041	5.080941727	3.149022536		

(d)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	2	8	256
	count	time	saving			
LMN	362797056	1946.209	1			
f_1SFMN (THEORY)	22674816	121.6380625	16	Difference		
f_1SFMN (REAL)	22571136	1089.717	1.785976543	8.958684293		

Figure 6.8: FFBP Test Results, simulation scene (A), focused scene (II), Batch 3

(a)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	1296	1	1296
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	1	0.333333333	0.5	1	10	1
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	1296	1	1	432	648	1
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	1296	1296	1296	0.333333333	0.5	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	1296	279936	362797056	2176782336	0.166666667	362797056
Instruction Count	LMN	f_lSFMN	f_lFMN	SFNM	FNM	eln(L)NM
	362797056	362797056	362797056	362797056	362797056	5453721.221
	count	time	saving			
LMN	362797056	1946.209	1			
f_lSFMN (THEORY)	362797056	1946.209	1	Difference		
f_lSFMN (REAL)	362797056	1946.209	1	1		

(b)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	3	5	243
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	5.333333333	1.777777778	2.666666667	5	10	5
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	3	81	81	5.333333333	8	432
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	16	3	3	144	216	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	104976	42.66666667	4478976	144	31104	4478976
Instruction Count	LMN	f_lSFMN	f_lFMN	SFNM	FNM	eln(L)NM
	362797056	22394880	4478976	4199040	839808	5453721.221
	count	time	saving			
LMN	362797056	1946.209	1			
f_lSFMN (THEORY)	22394880	120.136358	16.2	Difference		
f_lSFMN (REAL)	22394880	440.844	4.41473401	3.669530251		

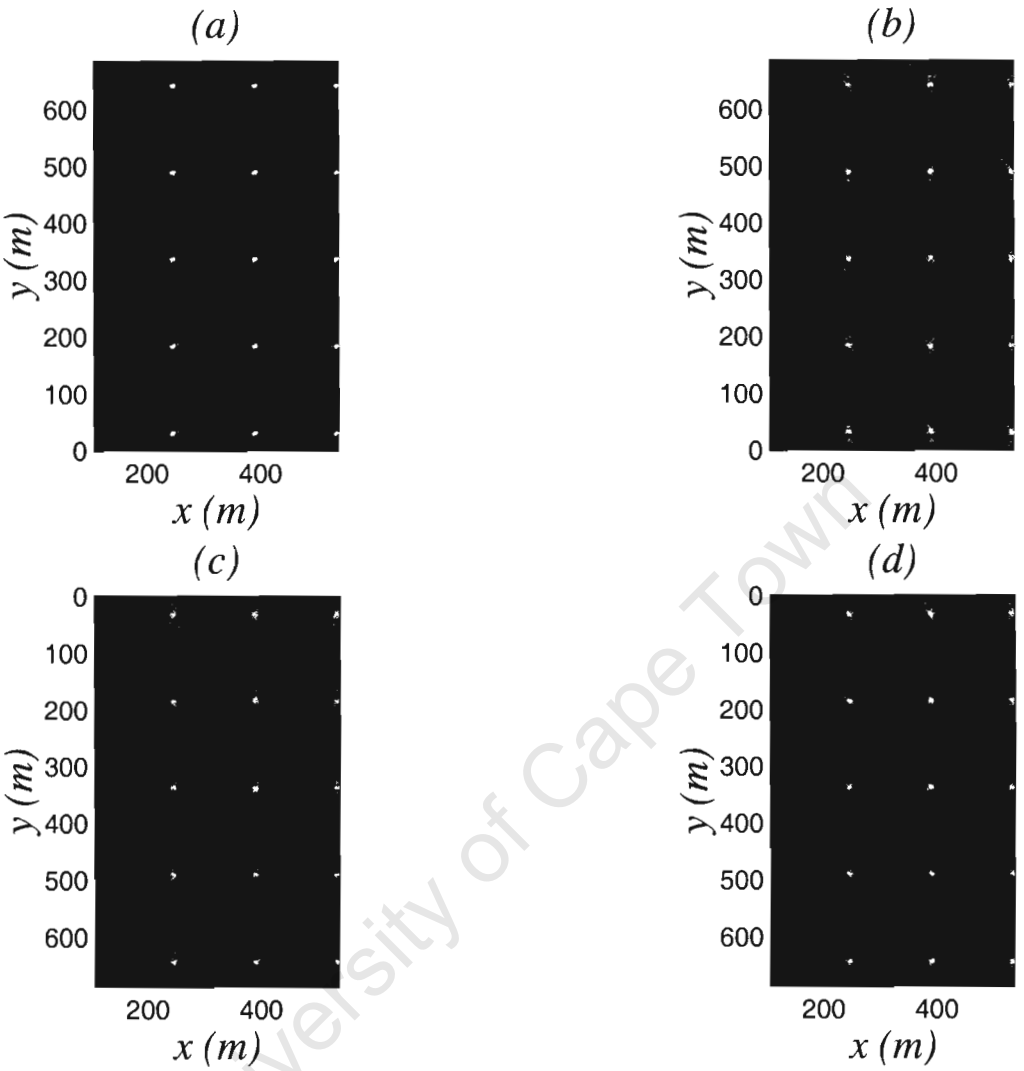
(c)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	4	4	256
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	5.0625	1.6875	2.53125	4	10	4
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	4	64	64	6.75	10.125	324
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	20.25	4	4	108	162	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	82944	68.34375	5668704	324	17496	5668704
Instruction Count	LMN	f_lSFMN	f_lFMN	SFNM	FNM	eln(L)NM
	362797056	22674816	5668704	4478976	1119744	5453721.221
	count	time	saving			
LMN	362797056	1946.209	1			
f_lSFMN (THEORY)	22674816	121.6380625	16	Difference		
f_lSFMN (REAL)	22605696	383.041	5.080941727	3.149022536		

(d)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	2	8	256
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	5.0625	1.6875	2.53125	8	10	8
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	2	128	128	3.375	5.0625	648
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	10.125	2	2	216	324	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	165888	17.0859375	2834352	40.5	69984	2834352
Instruction Count	LMN	f_lSFMN	f_lFMN	SFNM	FNM	eln(L)NM
	362797056	22674816	2834352	4478976	559872	5453721.221
	count	time	saving			
LMN	362797056	1946.209	1			
f_lSFMN (THEORY)	22674816	121.6380625	16	Difference		
f_lSFMN (REAL)	22571136	1089.717	1.785976543	8.958684293		

Figure 6.9: Information of FFBP Test Results, simulation scene (A), focused scene (II), Batch 3



(a)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	1296	1	1296
	count	time	saving			
LMN	362797056	1946.209	1			
f_1 SFMN (THEORY)	362797056	1946.209	1	Difference		
f_1 SFMN (REAL)	362797056	1946.209	1	1		

(b)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	12	3	1728
	count	time	saving			
LMN	362797056	1946.209	1			
f_1 SFMN (THEORY)	7558272	40.54602083	48	Difference		
f_1 SFMN (REAL)	6998400	46.177	42.14671806	1.138878712		

(c)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	11	3	1331
	count	time	saving			
LMN	362797056	1946.209	1			
f_1 SFMN (THEORY)	8994968.331	48.2531157	40.33333333	Difference		
f_1 SFMN (REAL)	84728176	60.457	32.1916238	1.252913913		

(d)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	10	3	1000
	count	time	saving			
LMN	362797056	1946.209	1			
f_1 SFMN (THEORY)	10883911.68	58.38627	33.33333333	Difference		
f_1 SFMN (REAL)	10352448	74.477	26.13167824	1.275590991		

Figure 6.10: FFBP Test Results, simulation scene (A), focused scene (II), Batch 4

(a)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	1296	1	1296
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	1	0.333333333	0.5	1	10	1
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	1296	1	1	432	648	1
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	1296	1296	1296	0.333333333	0.5	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	1296	279936	362797056	2176782336	0.166666667	362797056
Instruction Count	LMN	f_l SFMN	f_l FMN	SFNM	FNM	eln(L)NM
	362797056	362797056	362797056	362797056	362797056	5453721.221
	count	time	saving			
LMN	362797056	1946.209	1			
f_l SFMN (THEORY)	362797056	1946.209	1	Difference		
f_l SFMN (REAL)	362797056	1946.209	1	1		

(b)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	12	3	1728
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	0.75	0.25	0.375	3	10	3
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	12	144	144	3	4.5	108
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	9	12	12	36	54	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	186624	13.5	2519424	1296	1944	2519424
Instruction Count	LMN	f_l SFMN	f_l FMN	SFNM	FNM	eln(L)NM
	362797056	7558272	2519424	10077696	3359232	5453721.221
	count	time	saving			
LMN	362797056	1946.209	1			
f_l SFMN (THEORY)	7558272	40.54602083	48	Difference		
f_l SFMN (REAL)	6998400	46.177	42.14671806	1.138878712		

(c)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	11	3	1331
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	0.973703982	0.324567994	0.486851991	3	10	3
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	11	121	121	3.570247934	5.355371901	117.8181818
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	10.7107438	11	11	39.27272727	58.90909091	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	156816	19.12000546	2998322.777	1296	2313.520661	2998322.777
Instruction Count	LMN	f_l SFMN	f_l FMN	SFNM	FNM	eln(L)NM
	362797056	8994968.331	2998322.777	9237888	3079296	5453721.221
	count	time	saving			
LMN	362797056	1946.209	1			
f_l SFMN (THEORY)	8994968.331	48.2531157	40.33333333	Difference		
f_l SFMN (REAL)	84728176	60.457	32.1916238	1.252913913		

(d)

Input	L	M	N	F	S	$L_S=F^S$
	1296	432	648	10	3	1000
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	1.296	0.432	0.648	3	10	3
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	10	100	100	4.32	6.48	129.6
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	12.96	10	10	43.2	64.8	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	129600	27.9936	3627970.56	1296	2799.36	3627970.56
Instruction Count	LMN	f_l SFMN	f_l FMN	SFNM	FNM	eln(L)NM
	362797056	10883911.68	3627970.56	8398080	2799360	5453721.221
	count	time	saving			
LMN	362797056	1946.209	1			
f_l SFMN (THEORY)	10883911.68	58.38627	33.33333333	Difference		
f_l SFMN (REAL)	10352448	74.477	26.13167824	1.275590991		

Figure 6.11: Information of FFBP Test Results, simulation scene (A), focused scene (II), Batch 4

theoretically estimated value. This leads to a second problem of recursive function calling which slows down the algorithm as more stages are used. Every new FFBP algorithm which is called must have its own function space with hundreds of control variables which has to be reinitialised to process the scene. As more stages are used the number of images increase exponentially resulting in the extra time to declaration and initialisation to the control variable. This is in addition to the amount of intermediate data storage space which is also time consuming to manage.

One thing should be noted about the implementation of the FFBP algorithm done for the thesis project. This implementation assumes the worst case scenario by setting all new downrange profiles to O instead of $O_{MN}[s]$ in order to ensure accurate ground range focusing. By relaxing this constraint, it will allow much less data management and faster management of storage resources. Thus any optimisation done to the intermediated data storage in this respect will yield faster processing times.

What can be seen from the tests is that the more stages which are used when $L \neq F^S$, the slower the FFBP focusing becomes and the more the speed of processing deviates from the predicted value. Data movement and the creation of memory and temporary files, for the disk arrays, as well as the freeing and deletion of these resources needed in the intermediate stages of processing, all require time. However the instruction count does not take into account these non-theoretical practical tasks. Since the amount of sub-images increases exponential when using more stages to process an image, so the amount of resources which must be managed in this way, also increases. Thus there is a drop in the efficiency of the algorithm. This can be attributed to the fact that storage space for data in a PC must be created and destroyed whether for declared data or function calls. The relation $L = F^S$ or $L \approx F^S$ on the other hand maintains a balanced ratio of sub-aperture points to sub-image points thus ensuring a minimum amount of data movement and intermediate storage to ensuring times which are quite close time to the estimated FFBP processing time.

6.3 Real Scene Focusing

In addition to the tests done on simulated data, a real scene[2] was also processed. However the raw SAR data was not obtained from a wide beam radar and in theory is not the best source for unfocused SAR data when focusing using FFBP[3].

The setup information to focus the scene is given below in Tables 6.10 to 6.15. Table 6.10 gives the pulse information, Table 6.11 gives flight path information, Tables 6.12 gives the digital elevation information and table 6.13 gives the antenna beam pattern information. Table 6.14 gives the information regarding the unfocused SAR data and table 6.15 gives the information of the ground range focusing grid.

[2]The data was obtained from the CSIR Roof SAR system. The SAR system is used for SAR testing by the CSIR in Tswane (Pretoria). The system consists of a radar mounted on a 100 meter long horizontal beam at about 20 meters from the ground. The radar looks down a slope which drops about 25 meters every 300 meters. Bright targets in the scene are usually trees, some small buildings and a fence in the image. However the radar has a narrow, 2.7^0, beamwidth.

[3]The wider the beamwidth $\Delta\upsilon$ the more accurate backprojection is as more information is used to focus a single point on the image. Narrow beam data may have beam widths that are much smaller than the sub-apertures. Each set of downrange profiles corresponding to a sub aperture will form a new downrange profile for processing in the next stage. If the beamwidth $\Delta\upsilon$ is smaller than the sub-apertures then only one downrange profile of the newly formed set of downrange profiles will contain relevant information to focus the data for a particular point. Thus only one point along the range loci would contain information for subsequent stages of processing. This makes the backprojection less accurate as the method of coherent summing becomes less effective.

Description	Dimensions	Value	Unit
Wave speed	c	$3e8$	$[ms^{-1}]$
Voltage	V_t	1	$[V]$
Centre frequency	f_c	9.5	$[GHz]$
Bandwidth	Δf	1.2075	$[GHz]$

Table 6.10: Transmitted signal waveform information for test scenes (III)

Description	Dimensions	Value	Unit
RFP Resolution (Average)	dl	0.084814	(m)
RFP Vector Length	L	1238	(l)
Squint Angle	θ_{Squint}	$0(0^0)$	$(rad\,(^0))$
Depression Angle	$\phi_{Depression}$	$0.040556(0.12741^0)$	$(rad\,(^0))$

Table 6.11: Radar flight path information for test scene (III)

Description	Dimensions	Value	Unit
DEM Centre	$(X_I \times Y_J)$	371.96×105	$(m \times m)$
DEM Dimensions	$(\Delta x \times \Delta y)$	$185.89.94 \times 105$	$(m \times m)$
DEM Resolution	$(dy \times dx)$	0.1312×35	$(m \times m)$
DEM Image Size	$[I_X \times J_Y]$	2835×3	$[i_x \times j_j]$

Table 6.12: Parameters for DEM footprint for test scene (III)

Description	Dimensions	Value	Unit
ABP Centre	$(\Theta_A \times \Phi_E)$	0×0	$(rad \times rad)$
ABP Dimensions	$(\Delta\theta_A \times \Delta\phi_E)$	$2\pi \times \pi$	$(rad \times rad)$
ABP Resolution	$(d\theta_A \times d\phi_E)$	$\frac{\pi}{512} \times \frac{\pi}{512}$	$(rad \times rad)$
ABP Image Size	$[I_\Theta \times J_\Phi]$	1024×512	$[i_\theta \times j_\phi]$

Table 6.13: Parameters for antenna beam pattern footprint, test scene (III)

Description	Value	Unit	Dimensions
USD Start	$(s_A \times R_I)$	0×0	$(m \times m)$
USD Dimensions	$(\Delta s_A \times \Delta R)$	105×1500	$(m \times m)$
USD Resolution	$(ds_A \times dR)$	$\frac{105}{1238} \times \frac{1500}{12200}$	$(m \times m)$
USD Image Size	$[L \times O]$	1232×12200	$[l \times o]$

Table 6.14: Parameters for unfocused SAR data, test scene (III)

Description	Value	Unit	Dimensions
FSD Centre	$(X_C \times Y_C)$	-57.5×157.5	$(m \times m)$
FSD Dimensions	$(\Delta x_m \times \Delta y_n)$	105×315	$(m \times m)$
FSD resolution	$(dx_m \times dy_n)$	$\frac{1}{9} \times \frac{1}{9}$	$(m \times m)$
FSD Image Size	$[M \times N]$	945×2835	$[m \times n]$

Table 6.15: Parameters for focused SAR data ground range image grid, test scene (III)

A few more useful variables are given in 6.16.

Description	Value	Unit
Flight Path Length	105	$[m]$
Swath Beam Width	2.7	$[^0]$
Elevation Beam Width	2.7	$[^0]$
DEM Centre Height	48	$[m]$

Table 6.16: Extra scene (III) information

Figure 6.12 shows the results of focusing a real scene. Figure 6.12 (a) shows the unfocused raw SAR data, Figure 6.12 (b) shows the focusing of the scene done by an independent time domain focusing processor[4]. In addition the processor Figure 6.12 (c) shows the focusing of a scene by the FBP focusing processor and Figure 6.12 (d) shows the focusing of the scene by the FFBP processor. The factorisation information is given in Table 6.13.

The scene was acquired from a completely linear aperture. Figure 6.12 (d) was focused in slant range and Figure 6.12 (c) & (d) where focused with a DEM which was traced out by the radar boresight simulating slant range focusing. The scene was focused using an antenna beam pattern with swath and elevation parameters given in Table 6.16.

The results in Figure 6.12 were enhanced by clipping. Figure 6.12 (a), (b), (c) & (b) only shows a 16^{th} of the original maximum value of the focused image. The three bright targets in the near field in are quite bright and require that only a 16 of original maximum value of the focused image is required to see other detail.

Note that the FBP and FFBP algorithms are not narrow beam narrow band algorithms, however the result produced for the FBP processing still seem acceptable. The FFBP result however fragments and does not give a smooth focused image.

A test was also done to see how FBP will cope with focusing SAR data capture using a radar with a narrow beam without the aid of the antenna beam pattern. The result show in Figure 6.14, shows that FBP could not focus a scene without the antenna beam pattern as in Figure 6.3 (c) however in Figure 6.3 (b) a antenna beam pattern is not needed due to the wide beam width.

6.4 Multi-Buffer Tests

A short test was also performed to see the difference of processing data using the multi-buffer disk array by storing data on disk or by storing data in memory and relying on the OS page memory management to ensure efficient processing. The test shows that as array in memory becomes larger the efficiency of sequential reads and writes decreases, whereas the multi buffer disk array still is consistent with its speed of access.

The main problem with both methods of array implementation is the huge amount of time that it takes an array to resize. However the implementation of the algorithms above does not require any resizing of any array. If the resized column is ignored it can clearly be seen that the multi-buffer disk approximately is only 1.5 times slower when indexing directly from memory when the size of an array increases to $16M$. The time taken to destroy the disk array is faster for large arrays

[4]The independent time domain SAR processor that was used forms part of the G2 processor. G2 processor was developed at UCT to focus real SAR data.

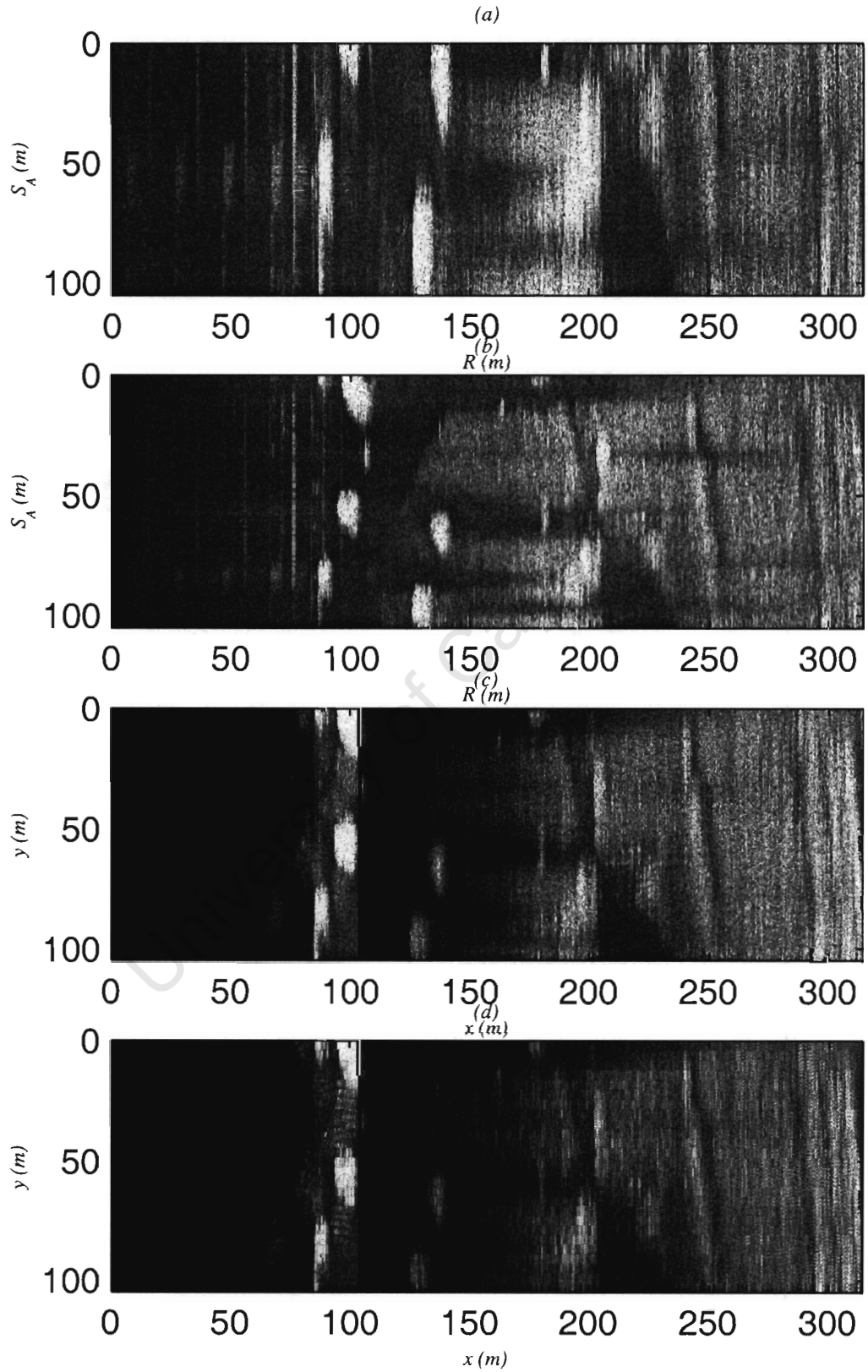


Figure 6.12: Real scene, test scene (III)
(a) shows the unfocused raw SAR data, (b) shows the focusing of the scene done by a independent time domain focusing processor, (c) shows the focusing of a scene by the FBP focusing processor and (d) shows the focusing of the scene by the FFBP processor.

Input	L	M	N	F	S	$L_S=F^S$
	1238	2835	945	35.1852242	2	1238
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	1	2 165/569	358/469	2	10	2
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	35.18522417	35.18522417	35.18522417	80.5735949	26.85786498	35.18522417
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	35.18522417	35.18522417	35.18522417	80 265/462	26 169/197	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	43559.30752	2164.034733	94263854.45	43559.3075	2164.034733	94263854.45
Instruction Count	LMN	f_l SFMN	f_l FMN	SFNM	FNM	eln(L)NM
	3316694850	188527708.9	94263854.45	188527709	5358150	51860384.9
	count	time	saving			
LMN	3316694850	15892.432	1			
f_l SFMN (THEORY)	188527708.9	903.3582917	17.59261209	Difference		
f_l SFMN (REAL)	183981210	1039.795	15.28419737	1.15103277		

Figure 6.13: Real scene factorisation information,test scene (III)

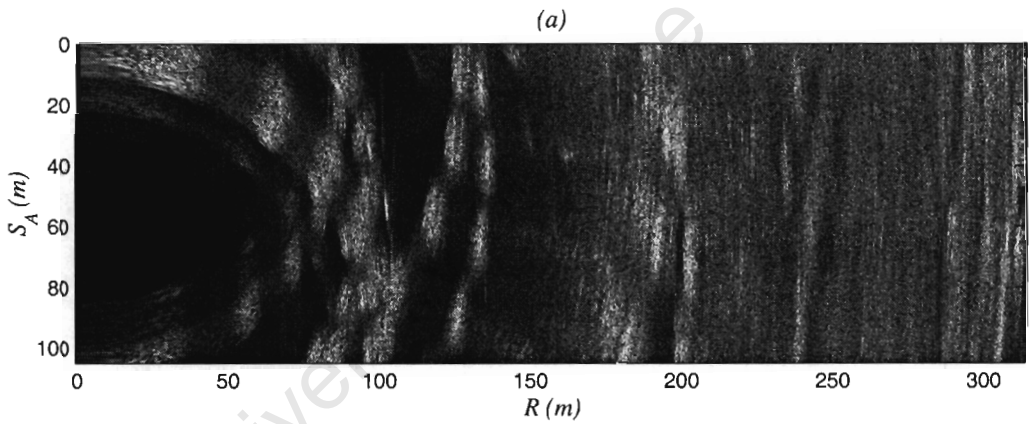


Figure 6.14: FBP focusing of a narrow beam unfocused SAR data without antenna beam pattern, test scene (III)

Input	L	M	N	F	S	$L_S=F^S$
	1238	2835	945	1238	1	1238
Constants	f_l	f_m	f_n	S_F	S_L	S_{MN}
	1	2 194/669	358/469	1	10	1
Inital Stage	$L_F[1]=F$	$M_F[1](O_{MN}[1])$	$N_F[1]$	$I_M[1]=Ff_m$	$J_N[1]=Ff_n$	$K_L[1]$
	1238	1	1	2835	945	1
Final Stage	$L_F[S]=Ff_l$	$M_F[S]=F$	$N_F[S]=F$	$I_F[S]$	$J_F[S]$	$K_F[S]$
(Sub-Image)	1238	1238	1238	2 194/669	358/469	1
Instruction Count	$L_F xK_L xO_{MN}$	$I_M xJ_N$	$L_F xK_L xO_{MN} xI_M xJ_N$	$Fxf_l xFxF$	$I_F xJ_F$	$Fxf_l xFxFxI_F xJ_F$
Initial - Final	1238	2679075	3316694850	1897413272	1.74800867	3316694850
Instruction Count	LMN	f_l SFMN	f_l FMN	SFNM	FNM	eln(L)NM
	3316694850	3316694850	3316694850	3316694850	2679075	51860384.9
	count	time	saving			
LMN	3316694850	9996.334	1			
f_l SFMN (THEORY)	3316694850	9996.334	1	Difference		
f_l SFMN (REAL)	183981210	9996.334	1	1		

Figure 6.15: Real scene factorisation information test scene (III)

since it only requires the deleting of a file. From the table it is clear that it is better to use large arrays on disk and smaller arrays in memory. The multi-buffer disk array has been implemented to limit the amount of storage which an array can have in memory. If the array exceeds the limit then the array is s stored on disk.

The only arrays which are accessed exclusively from disk are the arrays which are opened by giving a file name like the input data in all of the processing examples. The results can be seen in Figure 6.16.

(a)

	Length	(Size)								Time	(Duration)			
	Max Mem	Inital Array	Resized Array	Sectors	Sector	Buffer	Mem Buf	Type	Physical	Creation	Resizing	Read-Write	Destroying	Total
1	1	256	1024	1	32	32	32	16	16384	0	0	0	0	0
2	1	1024	4096	4	32	32	128	16	65536	0	0	0	0.01	0.01
3	1	4096	16384	16	32	32	512	16	262144	0	0	0.02	0	0.02
4	1	16384	65536	64	32	32	2048	16	1048576	0	0.03	0.07	0	0.1
5	1	65536	262144	256	32	32	8192	16	4194304	0.01	0.12	0.23	0.01	0.37
6	1	262144	1048576	1024	32	32	32768	16	16777216	0.05	1.573	1.261	0.691	3.575
7	1	1048576	4194304	4096	32	32	131072	16	67108864	1.012	43.322	28.06	4.166	76.56
8	1	4194304	16777216	16384	32	32	524288	16	268435456	6.7	413.725	164.577	55.67	640.672

(b)

	Length	(Size)								Time	(Duration)			
	Max Mem	Inital Array	Resized Array	Sectors	Sector	Buffer	Mem Buf	Type	Physical	Creation	Resizing	Read-Write	Destroying	Total
1	1073741824	256	1024	0	0	0	0	16	16384	0	0	0	0	0
2	1073741824	1024	4096	0	0	0	0	16	65536	0	0	0	0	0
3	1073741824	4096	16384	0	0	0	0	16	262144	0	0.01	0	0	0.01
4	1073741824	16384	65536	0	0	0	0	16	1048576	0	0.03	0.01	0.01	0.05
5	1073741824	65536	262144	0	0	0	0	16	4194304	0.01	0.13	0.04	0.05	0.23
6	1073741824	262144	1048576	0	0	0	0	16	16777216	0.06	0.491	0.17	0.16	0.881
7	1073741824	1048576	4194304	0	0	0	0	16	67108864	0.231	28.671	0.671	13.7	43.273
8	1073741824	4194304	16777216	0	0	0	0	16	268435456	1.902	477.016	104.851	336.334	920.103

Figure 6.16: Disk array and memory Array Comparison

Chapter 7

Conclusions and Recommendations

In pursuit of investigating algorithms which drastically reduce the amount of processing time, the time domain multi-stage Fast Factorised Filter Backprojection SAR inversion method (FFBP) explained in [1] and [2] was investigated and then implemented. The SAR digital image reconstruction processor was designed to focus images to ground range and take into account a general flight path, uneven ground and antenna beam pattern.

The tests in Section 6.2.2 which compared the time taken to digitally reconstruct an image using FBP to the time taken to digitally reconstruct an image using FFBP showed that there is a vast increase in time saved when performing SAR inversion using the FFBP method. However the increase is most prevalent when ensuring that the number of aperture points, factor and amount of stages used are related by $L = F^S$. Deviating from this theoretical optimum results in a deviation from the theoretically predicted saving. Huge deviation in some cases makes the FFBP algorithm an inefficient choice for processing data since the time taken to complete focusing is longer than some cases using FBP.

Spectral domain algorithms like the Range Doppler algorithm are suited to focus SAR data acquired by a straight flight path using a narrow band SAR system with radar which produces a narrow beam. In contrast the time domain SAR focusing using filtered backprojection is well suited to producing images from ultra wide band SAR data acquired along a general curved flight path using a radar that produces a wide beam. Thus the FFBP provides an alternative fast SAR processing algorithm that can complement the processing of SAR images in the spectral domain.

In theory, filtered backprojection can completely recover the scene from data with an infinite bandwidth acquired using an isotropic antenna along a generally curved flight path of infinite length. Thus the accuracy of a focused image using filtered backprojection is only limited by a finite bandwidth, non isotropic beam width and a finite flight path length. This means that filtered backprojection approximates the theoretically ideal case of processing data of infinite bandwidth acquired using an isotropic antenna as well as the practical case of data being acquired along a arbitrary curved flight path. Since the FFBP algorithm combines the best of both worlds giving an algorithm that is suited to processing ultra wide band SAR data acquired along a general curved flight path using a radar that produces a wide beam while still having an instruction count that is proportional on $\mathcal{O}\left(n^2 \log\left(n\right)\right)$ process, it is concluded that the FFBP algorithm is highly recommended for fast and accurate processing of SAR data to produce high quality digitally reconstructed images.

In South Africa, the South African Synthetic Aperture Radar(SASAR 1) project project ran in the

late 1990s from between 1996 - 1997 [16] and collected SAR data from various parts of South using a VHF SAR system that operated at a low centre frequencies [17]. The design of SASAR 1 was similar to the Coherent All RAdio BAnd Sending (CARABAS) SAR system used by the FOA [3, 16]. However FFBP is a recently developed algorithm that was is not available at the time of designing SASAR 1. The earliest known publication used in the thesis was published in 2000. [2] Since there have been projects in the pass that would of benefit from the type of focusing that is discussed in this thesis, it is a good investment for future research at UCT to investigate the FFBP algorithm.

7.1 Chapter Summary

Chapter 2 developed a radar echo modal which would not only take into account a straight flight path but be representative of unfocused SAR data received from a general flight path. The model was capable of representing the signal received from any point in the scene using an arbitrary beam pattern. The model is however restricted to represent range compressed SAR data down converted to baseband.

Chapter 3 explained the concept of exact inversion and showed that filtered backprojection used in SAR is an approximation of this method. A mathematical model for ground range focusing was presented which would form the bases of the digital image reconstruction in this thesis.

Chapter 4 analysed the the digital image reconstruction FBP and FFBP method. Important concepts relating to the theory were discussed here. Detailed examples and detailed formulae and theory was also presented in this chapter. The chapter intended giving the reader a complete understanding of FFBP from a theoretical point of view.

Chapter 5 presented practical issues relating to implementation problems and highlighted theoretical constraints present in the implementation of the FFBP algorithm. The FFBP was modified so that the factorisation process can be more tolerant of the relationships between the dimensions of input data and output data. The implementation of the multi-buffer disk array which ensures better throughput was presented. A general instruction count for the implemented FFBP algorithm was also derived.

Chapter 6 showed that the although the FFBP is indeed faster than the FBP the data movement in the PC does affect the efficiency of the algorithm. It is concluded that if $L = F^S$ or $L \approx F^S$ that the decrease in new aperture points (L_K) balance, out the increase in sub-images $(I_F \times J_F)$ resulting in the factorisation process requiring less storage resources and recursive function calls thus ensuring the best saving next to the theoretically estimated processing time.

7.2 Future Work

Due to the recursive partitioning of images into separate focusing problems SAR inversion using FFBP can also be extended to make efficient use of parallel processing. The 2-stage version has been implemented as a real time algorithm on a parallel architecture which is explained in [3]. It is also suggested in [11] that the implementation of FFBP can also be used as a real time algorithm on a parallel architecture. This implementation has been implemented to process sub-images in a preordered manner with the intention of implementing the parallel architecture. In theory all that

is needed is to recursively call sub images as separate processes on a multi processor machine to enable parallel implementation of the FFBP algorithm.

In theory a $2D$ aperture can focus a $3D$ block of data. It may also be worth investigating the possibility of using the algorithm in the field of ground radar to produce a $3D$ block of the ground below the ground surface.

In [8] it can be seen that the unfocused SAR data is circular band limited. It is proven that hexagonal sampling is the best sampling scheme for representing circular band limited signals [20] giving the most bandwidth to samples ratio [20]. Since equation 3.5 shows that data in one range bin comes from a circular locus with radar at the centre, it may be worth looking into performing a mathematical study to determine if focusing to a hexagonal grid would help to preserve the accuracy of a final focused image.

The flight path model of the radar echo model can be extended to keep track of azimuth, elevation and tilt (polarisation) angle of the radar at every sample point along the flight path thus extending the model to spotlight and scan SAR mode. Also the velocity and acceleration vectors can be added to accommodate the changing angle of the beam pattern around sharp corners. If the fight path $\vec{u}(t_A)$ can be made to represent the flight path of a sending radar, another flight path $\vec{w}(t_A)$ can be added to represent flight path of a receiving radar. The model can then be modified to handle bistatic SAR. The modal can also be extended to represent Inverse SAR, in which the target moves past the sensor on a path $\vec{v}(t_A)$.

A more in depth quantitative analysis of focusing performance on focused impulse response could also be investigated in terms of width and shape of main peak, sidelobe levels and phase errors across the peak.

At UCT a frequency domain SAR focusing processor called the G2 processor that uses the *Range Doppler* algorithm for SAR inversion has already been developed. This processor was built to focus real SAR data. The Range Doppler algorithm is a Narrow Beam High Frequency SAR focusing algorithm in contrast to the FFBP which is a Wideband Low Frequency algorithm. Since this thesis project focused simulated SAR data it can serve as a platform to build a fast time domain focusing processor that can complement the G2 processor by using the FFBP algorithm to focus real SAR data.

Appendix A

The Radon Transform

The Radon transform can be used in N dimensional space, but for the purposes of this thesis the transform will only be described for 2 dimensions. A formal explanation of the Radon transform in $2D$ is as follows.

A.1 Radon Projection

Mathematically the Radon transform is represented by mapping, the value of, the result of each line integrals of a family of curves in the observed scene to a unique coordinate in the projected scene. The value of a line integral in the observed (object) scene would represent a point in the projected scene. Commonly, a family of straight lines that span the whole domain are used. The value of each of these line integrals would occupy a unique coordinate in the projected plane. The family of lines each produce a one-to-one mapping of the observed scene into the projected scene[15].

- In 2 dimensions it is the mapping of all possible straight line integrals passing through a scalar function $f(x, y)$ onto a projected space $g(r, \theta)$ [15].

Let R represent any straight line through the $x - y$ plane, then the backprojection integral can be defined by equationA.1 [15].

$$g(r, \theta) = \int_R f(x, y)\, dl \tag{A.1}$$

[15]

The integral transforms the information of the scalar function $f(x, y)$ on the $x - y$ plane to a scalar function $g(r, \theta)$ on the $r - \theta$ plane. By representing the equation of any straight line R on the $x - y$ plane by equations A.2 or A.3, below, r and θ can be varied to represent any line in the $x - y$ plane as can be seen in figure A.1.

$$r = x\cos(\theta) + y\sin(\theta) \tag{A.2}$$

$$= \langle x, y\rangle \bullet \langle \cos(\theta), \sin(\theta)\rangle \tag{A.3}$$

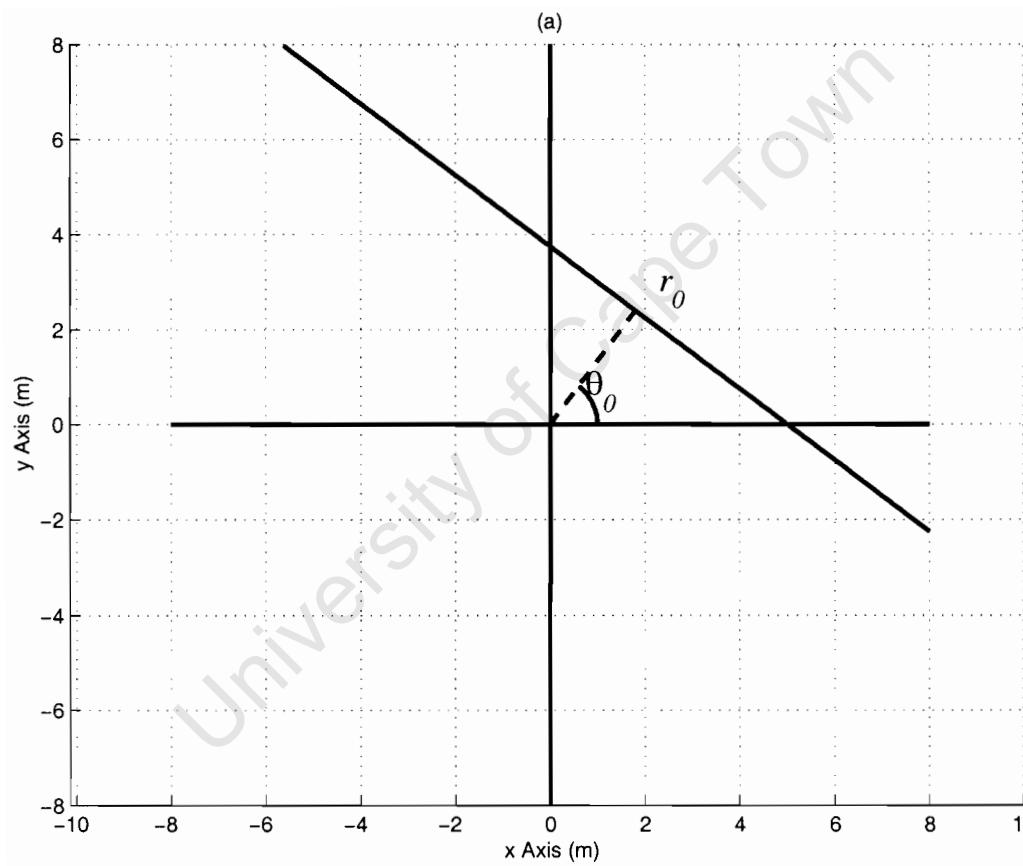


Figure A.1: The straight line R given by $r = x\cos(\theta) + y\sin(\theta)$

In the object scene the variables r and θ are constants. From equation A.3 it can be seen that the locus of the line R can also be traced out by the dot product between a constant vector $\langle \cos(\theta), \sin(\theta) \rangle$ and the vector $\langle x, y \rangle$ representing all points on the $x - y$ plane. The line integral can be expressed analytically by introducing the line L given by

$$l = y\cos(\theta) - x\sin(\theta) \tag{A.4}$$
$$= \langle x, y \rangle \bullet \langle \cos(\theta), -\sin(\theta) \rangle \tag{A.5}$$

The line L is always perpendicular to the line R . Thus these two lines can be represented by the following linear transformation:

$$\begin{aligned} r &= x\cos(\theta) + y\sin(\theta) \\ l &= y\cos(\theta) - x\sin(\theta) \end{aligned} \tag{A.6}$$

$$\begin{bmatrix} r \\ l \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{A.7}$$

By using the inverse of this transformation an analytical expression for the Radon transform can be derived from the following transform using the lines R and L:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} r \\ l \end{bmatrix} \tag{A.8}$$

[15]

Equation A.9 is produced by substitution of equation A.8 into equation A.1:

$$g(r, \theta) = \int f(r\cos(\theta) - l\sin(\theta), r\cos(\theta) + l\sin(\theta))\, dl \tag{A.9}$$

Figure A.2 shows that the transform of equation A.8 introduces the new coordinate system l and r. Figure A.3 shows the line R cutting through the 2D scalar function $f(x, y)$

An interpretation of the Radon transform is given in equations A.10 and A.11. In the projected scene the variables x and y are constants. The Radon transform for a function $f(x, y) = \delta(x - x_0, y - y_0)$ in the object (observed) scene would create a sinusoidal Dirac Delta in the projected plane. From equation A.9 the Radon transform of $f(x, y)$ is given by equations A.10 and A.11:

$$g(r, \theta) = \delta(x_0\cos(\theta) + y_0\sin(\theta) - r) \tag{A.10}$$
$$= \delta(r_0\cos(\theta - \theta_0) - r) \tag{A.11}$$

If the projected scene is represented in polar coordinates, the projection would be a circle with circumference r_0 and centre at $\langle \frac{r_0}{2}, \theta \rangle$. Figure A.4 gives the representation of three points on the $x - y$ plane and two representations of the Radon transform on different representations of the $r - \theta$ plane.

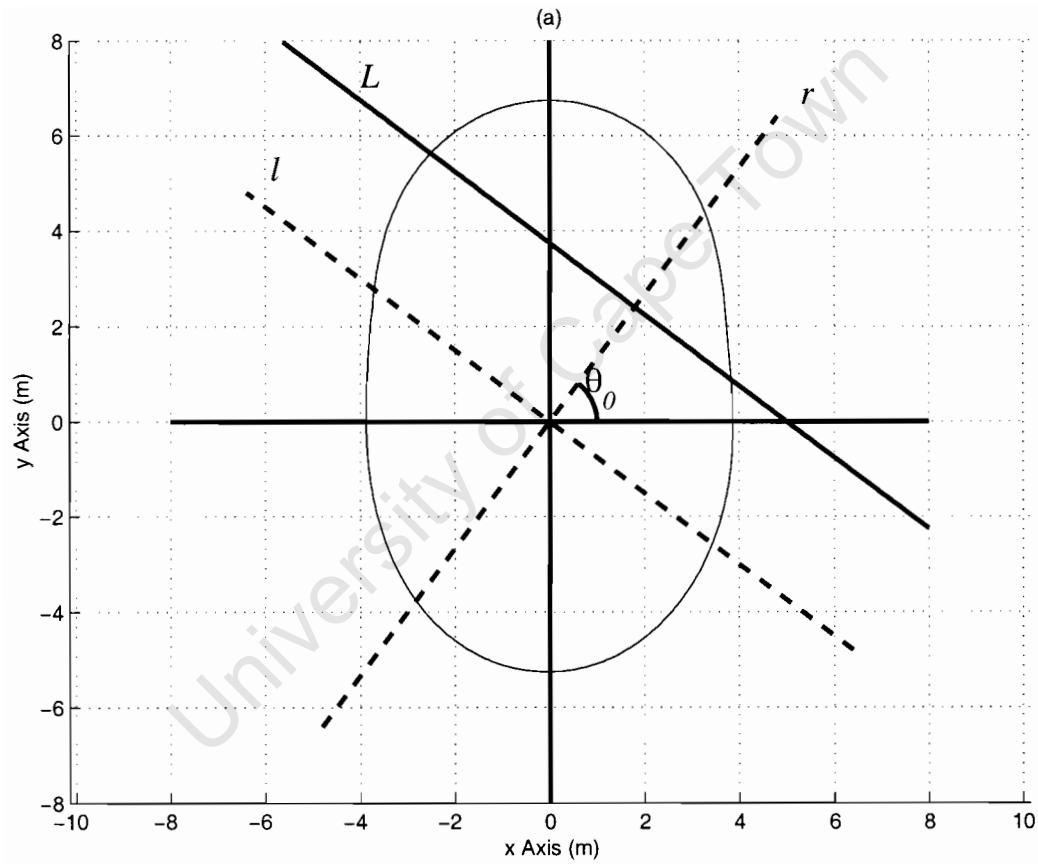


Figure A.2: Radon line integral

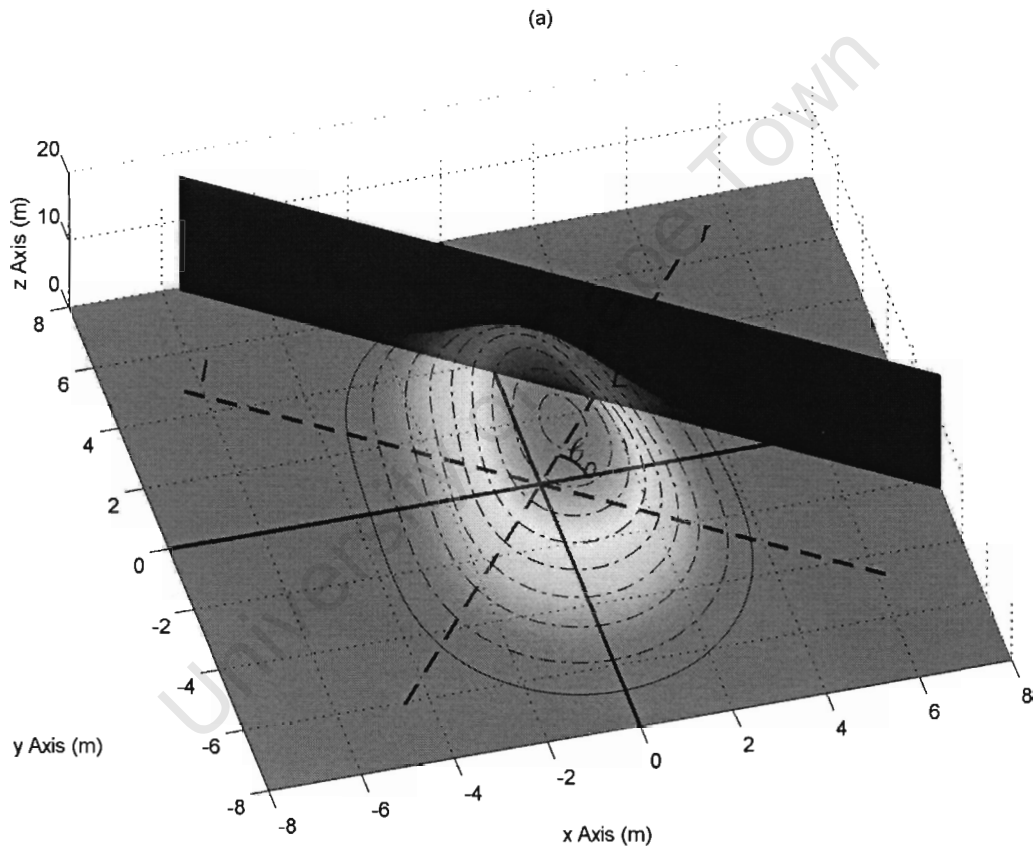


Figure A.3: Radon line integral (3D representation)

1. Figure A.4 (a) shows three footprints of Dirac Deltas on the $x - y$ plane.
2. Figure A.4 (b) shows the trace of the Radon transform of three Dirac sinusoids on a Cartesian $r - \theta$ plane.
3. Figure A.4 (c) shows the trace of the Radon transform of three Dirac sinusoids on a polar $r - \theta$ plane.

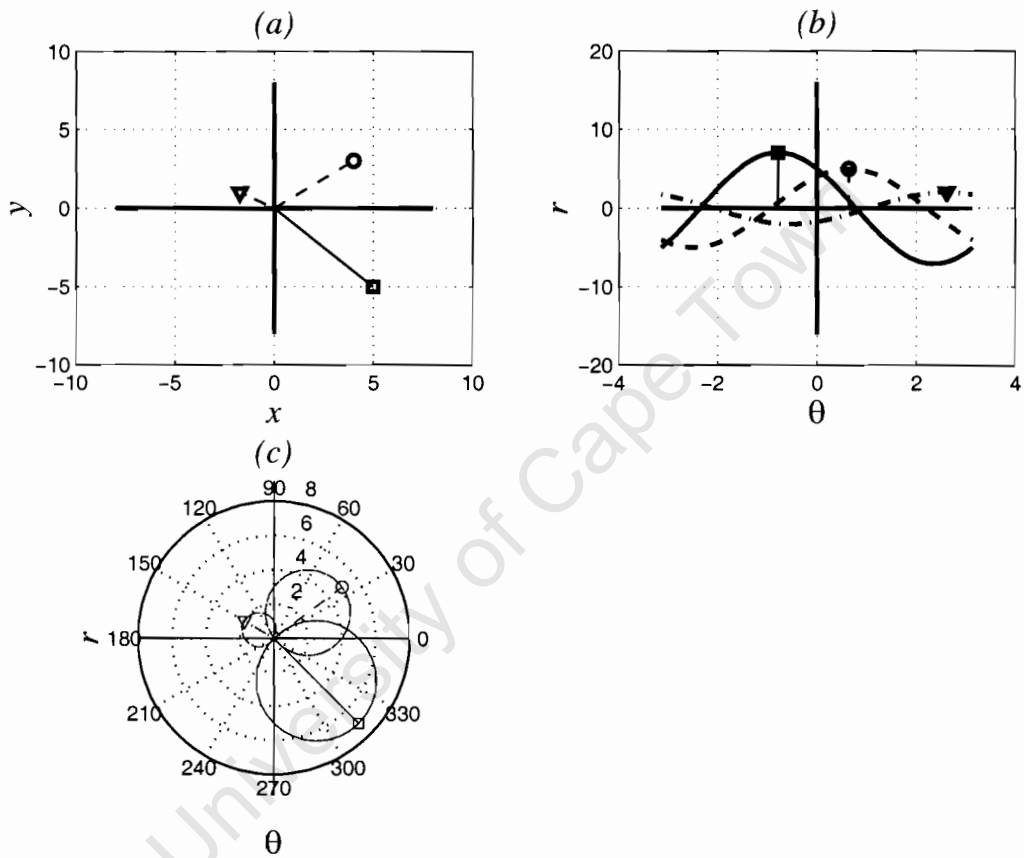


Figure A.4: Radon transform of three Dirac Deltas

A.2 Backprojection

An interpretation of the Inverse Radon transform is given in equations A.12. In the object scene the variables r and θ are constants. The Radon transform $g(r, \theta) = \delta(r - r_0, \theta - \theta_0)$ in the projected scene would create a line Dirac Delta in the object plane. From equation A.9 the Radon transform of $f(x, y)$ is given by equations A.10 and A.11:

$$f(x, y) = \delta(x \cos(\theta_0) + y \sin(\theta_0) - r_0) \tag{A.12}$$

If the projected scene is represented in polar coordinates, the projection would be a circle with circumference r_0 and centre at $\langle \frac{r_0}{2}, \theta \rangle$. Figure A.5 gives the representation of three points on the

$x - y$ plane and two representations of the Radon transform on different representations of the $r - \theta$ plane.

1. Figure A.5 (a) shows three footprints of Dirac Deltas on a Polar $r - \theta$ plane.
2. Figure A.5 (b) shows three footprints of Dirac Deltas on a on a Cartesian $r - \theta$ plane.
3. Figure A.5 (c) shows the line of the inverse Radon transform of three Dirac Deltas on the $x - y$ plane.

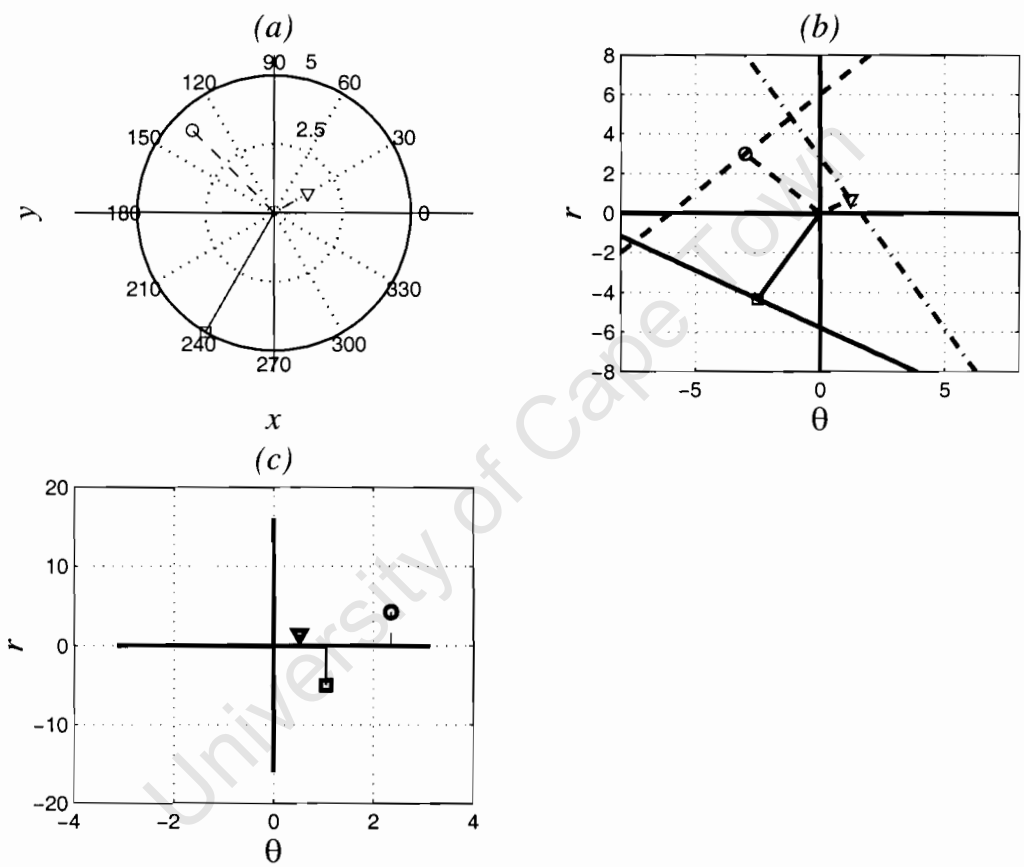


Figure A.5: Backprojection (Radon Inversion)

The back projected integral of $g(r, \theta)$ is given by equation A.13 [15]:

$$h(x,y) = \int_R g(r,\theta)\, d\theta \tag{A.13}$$

$$h(x,y) = \int_0^{\pi} g(x\cos(\theta) + y\sin(\theta), \theta)\, d\theta \tag{A.14}$$

$$= f(x,y) \otimes\otimes \frac{1}{r} \tag{A.15}$$

$$= f(x,y) \otimes\otimes \frac{1}{\sqrt{x^2+y^2}} \tag{A.16}$$

The $\frac{1}{r}$ convolution is caused by the integration of all possible curves of the form $r = x\sin(\theta) + y\cos(\theta)$ in the projected scene that passes through the equation $g(r,\theta) = \delta(x_0\cos(\theta) + y_0\sin(\theta) - r)$.

In order to overcome the inherent $\frac{1}{r}$ convolution and to recover the Dirac Delta at position $\langle x_0, y_0 \rangle$ the backprojection process has to be modified. This modification is called Filtered Backprojection and it can be shown that the image can be correctly recovered by passing the projected scene through a ramp filter before backprojection Firstly the function $g(r, \theta)$ is ramp filtering giving by

$$g'(r, \theta) = \mathcal{F}^{-1}(|\kappa| g(\kappa, \Omega))$$

where $\kappa \Leftrightarrow r$, $\Omega \Leftrightarrow \theta$ and $g(\kappa, \Omega) \Leftrightarrow g(r, \theta)$, and then it is backprojected to retrieve the original image given by :

$$f(x, y) = \int_R g'(r, \theta) \, d\theta \tag{A.17}$$

$$f(x, y) = \int_0^{\pi} g'(x \cos(\theta) + y \sin(\theta), \theta) \, d\theta \tag{A.18}$$

The frequency domain proof of the above method can be found in [15]. Figure A.6 gives a visual representation of the back projection process

1. Figure A.6 (a) & (b) shows the function $f(x, y)$
2. Figure A.6 (c) & (d) shows the radon projection $g(r, \theta)$ of the function $f(x, y)$
3. Figure A.6 (e) & (f) shows the backprojected function $g(r, \theta)$ producing the distribution $h(x, y)$
4. Figure A.6 (g) & (h) shows the backprojected function $g(r, \theta)$ producing the distribution $f(x, y)$ using filtered backprojection.

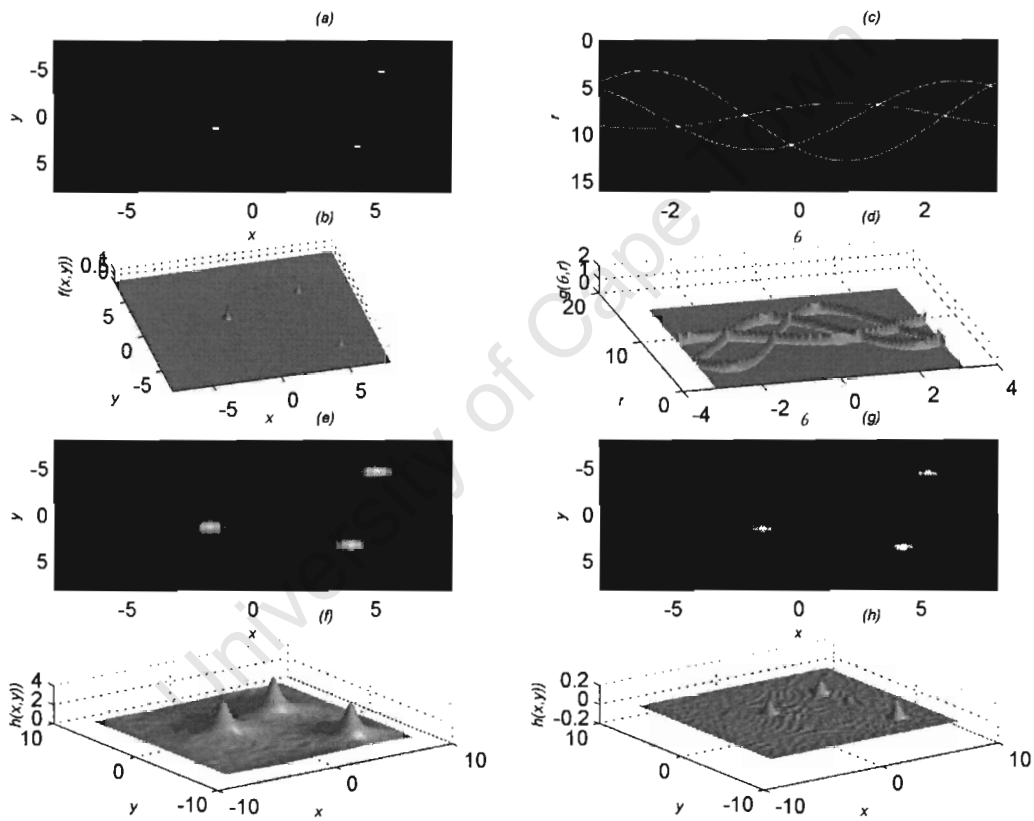


Figure A.6: Radon transform of three Dirac Deltas with 3D representation

Bibliography

[1] L.M.H, Ulander, H Hellsten, G.Stenstrom, Performance Analysis of Fast Backprojection for Synthetic Aperture Radar Processing, Proceedings of SPIE, vol 4382, pp 13-21, 2001

[2] L.M.H Ulander, H Hellsten, G.Stenstrom, Synthetic-Aperture Radar Processing using Fast Factorised Backprojection, Proceedings EUSAR '2000, Munich, Germany, 23-25 May 2000, pp 753-756, 2000

[3] O.Seger, M Herberthson & H . Hellsten, Real Time SAR Processing of Low Frequency Ultra Wide Band Radar Data, Proceedings EUSAR '98, Friedrichchafen, Germany, 25-27 May 1998, pp 489-492, 1998

[4] W.G.Carrara, R.S. Goodman & R.M. Majewski, Spotlight Synthetic Aperture Radar, Signal Processing Algorithms, Artech House, 1995

[5] C. Cafforio, C. Prati & F.Rocca, SAR Data Focusing Using Seismic Migration Techniques, IEEE, Trans Aerospace Electron Systems. 27, pp 194-207, 1991

[6] H. Hellsten & L.E. Anderson, An Inverse Method for the Processing of Synthetic Aperture Radar Data, Inverse Problems, 3, pp 111-124, 1987

[7] L.M.H. Ulander, Approaching the Wavelength Resolution Limit in Ultra-Wideband VHF-SAR, Proceedings of the PIERS Workshop on Advances in Radar Methods, Baveno, Italy, 20-22 July, pp 83-85, 1998

[8] L.M.H Ulander& H Hellsten, A new Formula for SAR Spatial Resolution, AUE Electronics and Communication, Vol 50, Part 2, pp 117-121, 1996

[9] J.A. Fawcett, Inversion of N-dimensional Spherical Means, SIAM Journal on Mathematical analysis, Vol. 45, pp 336-341, 1985

[10] L.E. Anderson, On the Determination of a Function from Spherical Averages, SIAM Journal on Mathematical analysis, Vol. 19, pp 214-232, 1988

[11] S.M Banks & H.D Griffiths, The use of fast factorised back projection for synthetic aperture imaging, Proceedings IOA Conference on Acoustics, Salford, England, 25-27 March, 2002

[12] A.J. Wilkinson, Notes on Radar Signal Processing, Ver. 6.1, June 2002

[13] S. Kingsley & S Quegan, Understanding Radar Systems, McGraw-Hill Book Company, 1992

[14] M. Soumekh, Synthetic Aperture Radar Signal Processing with Matlab Algorithms, John Wiley & Sons, 1999

[15] S.R Dean, The Radon Transform and some of its Applications, John Wiley & Sons, 1983

[16] M. da Silveira, J.M. Horrell , M.R. Inggs , E. Avenant, Progress on the SASAR System : First Results, International Geoscience and Remote Sensing Symposium 1997, International Geoscience and Remote Sensing Symposium 1997, Vol IV, pp 2015-2017, Aug 1997

[17] M da Silveira, External Polarimetric Calibration Considerations, CSIR - Manufacturing & Aeronautical Systems Technology, Num 7803-3836, pp 2018-2020, Jan 1997

[18] F.G. Stremler, Introduction to Communication Systems, Third Edition,1990

[19] Meetings with the Supervisor

[20] D.E. Dudgeon & R M Mersereau, Multidimensional Digital Signal Processing, Prentice-Hall, 1984

[21] M.A.Weiss, Algorithms, Data Structures and Problem Solving with C++, Addison Wesley Publishing Company, Inc, 1996.

[22] S.M. Banks & H.D. Griffiths, Imaging and Motion Estimation for Synthetic Aperture Sonar Based on Fast Factorised Back Projection

[23] M.I. Skolnix, Introduction to Radar Systems, McGraw-Hill, 1980

[24] AGARD, Fundamentals and Spacial Problems of Synthetic Aperture Radar, NATO, 1992

[25] D. Halliday, R Resnick, J Walker, Fundamentals of Physics Extended, Fourth Edition, John Wiley & Sons, 1993

[26] D.G. Zill & M.R Cullen, Advanced Engineering Mathematics, PWS Publishing Company, 1992

[27] R. Bamler, A Systematic Comparison of SAR Focussing Algorithms, Proc. of IGARSS '91, pp. 1005-1009, 1991

[28] B.C. Barber, Theory of Digital Imaging from Orbital Synthetic-aperture Radar, International Journal of Remote Sensing,Vol 6, Num 7, pp 1009-1057, Jan 1985